FSA Integration Partner Program
United States Department of Education
Office of Federal Student Aid

FSA Identity and Access Management Tools Analysis

Deliverable 143.1.3 Identity and Access Management Tools – Prototype

Version 2.0

May 28, 2004

Document Revision History

Version Number	Date	Author	Revisions Made
Version 1.0	May 14, 2004	Anu Sharma	Initial draft
Version 2.0	May 28, 2004	Jesse Bowen Anthony Ko Anu Sharma Ryan Summers	Sections: 4.3 4.4 4.7 4.10 6.2 6.3 Figures: 14 Appendices: A F G I

Table of Contents

Figures

1 Executive Summary

The goal of the prototype phase of the Identity and Access Management Tools Analysis Task Order was to provide FSA an opportunity to gain experience with installation and integration of COTS solutions for managing and administering access to FSA systems. This report documents the results of installation, testing, and integration of the IBM Tivoli Access Manager and Tivoli Identity Manager products in FSA development environments. Overall, the experience gained through this Proof of Concept effort provided valuable planning information.

Designs for installation environments were created to understand platform requirements and implications for larger-scale deployments. Design and build of code modifications for sample applications integrated with the Identity and Access Management tools will provide guidelines for future deployment of these capabilities to new and existing systems. Installation of the software at the FSA Virtual Data Center allowed identification of issues related to patch levels on the HP-UX 11i platform that are being addressed by the vendor. Installation of the vendor products in an on-site development environment allowed investigation and testing of component connectivity and functionality. In summary, these insights will substantially accelerate future deployment of Identity and Access Management technologies and aided understanding of how they can be integrated with FSA systems and applications to achieve FSA business objectives.

As a part of the prototype, FSA requested that the vendor install Tivoli Identity Manager and Tivoli Access Manager on the HP-UX 11i standard hardware configuration at the VDC. Several issues with Tivoli Access Manager and Tivoli Identity Manager running on HP-UX 11i prevented a complete installation during the on-site installation window. However, the prototype team continued the prototype effort by utilizing a pre-installed Windows version of Tivoli Access Manager provided by the vendor; TAM's functionality was successfully tested. The vendor also provided a VMWare version of Tivoli Identity Manager but the delay of working adapters to TAM or to provision the target application Oracle databases prevented significant testing of TIM. This deliverable documents the prototype phase and contains:

- A summary of the prototype tool business benefits to FSA
- A description of the Proof of Concept design
- The results of the Tivoli Access Manager and Tivoli Identity Manager installation efforts
- A summary of the level of effort to integrate the target applications with Tivoli Access Manager and Tivoli Identity Manager
- Test objectives and actual results
- A description of the major lessons learned and prospective next steps

2 Introduction

2.1 Background

The Identity & Access Management Tools Analysis task order was created to support FSA selection of appropriate technologies to support business objectives for improving security administration and access control capabilities for FSA's Trading Partners.

The first phase of this task order evaluated market leading vendor offerings for Web Access Control and Identity Management technologies. Next, a subset of COTS Identity Management and Web Access Control security tools were analyzed based on established evaluation criteria. The advantages and disadvantages of each product were documented and evaluated. After on-site software demonstrations, FSA decided to complete the Security Tools Proof of Concept with IBM's Tivoli Identity Manager and Tivoli Access Manager Products.

Deliverable 143.1.3 Identity & Access Management Tools – Prototype, the final deliverable of this task order, documents the installation, integration, and testing of the Proof of Concept in the FSA development environment. In addition, this document evaluates implications for future deployment and integration of these security tools at FSA.

2.2 Objectives

The overall objective of the Security Tools Proof of Concept was to install the Tivoli Identity Manager and Tivoli Access Manager components in a development environment and to integrate them with sample FSA applications. The development environment initially chosen was the FSA Integrated Technical Architecture (ITA) development environment in the FSA Virtual Data Center (VDC). The specific objectives of the Proof of Concept were to:

- Confirm security product compatibility with and support of FSA VDC standard hardware configuration (e.g. WAS on HP-UX 11i)
- Integrate Identity Management and Web Access Control tools with the FSA eZAudit application
- Test the Tivoli Identity Manager and Access Manager products against typical FSA user scenarios to gauge support for FSA business objectives
- Create a report that documents the above efforts and evaluates integration implications for future FSA use of these tools.

2.3 Approach

The Identity & Access Management Tools Analysis task order is divided into three major phases:

- Vendor Analysis Phase (Completed 1/23/04) – This phase established criteria for a vendor evaluation, identified market leading solutions, and selected products for on-site evaluation.
- Product Options Phase (Completed 3/12/04) – Conducted on-site vendor evaluation and testing, analyzed vendor solutions, and selected products for the prototype.
- Prototype Phase (Completed 5/14/04) – The team prototyped and tested the Identity Management and Web Access Control components in the FSA development environment against FSA business objectives.

To support the prototype effort, the project team:

- Organized several meetings with the vendor to coordinate vendor support of the Proof of Concept
- Attended weekly VDC development conference calls
- Met with the ACS Common Services for Borrowers team to discuss their experience with Tivoli Access Manager
- Drafted design documentation for product installation and application integration
- Set up discussions with IBM to discuss technical issues
- Worked with the VDC to request assistance on changes to the hardware for the security Proof of Concept
- Met with the ITA team to confirm use of the ITA Development environment for the prototype and finalize the design
- Assisted IBM with TIM and TAM product installation
- Integrated the target applications with TIM and TAM
- Conducted testing of product installations and application integration
- Documented impacts for FSA enterprise deployment

2.4 Document Overview

This deliverable summarizes the results of the Prototype phase of this project. Subsequent sections contain the following content:

Section 3 – Identity and Access Management solution description

Section 4 – Prototype Design and Implementation including installation

Section 5 – Summary of the Proof of Concept Application Integration efforts

Section 6 – Review of Proof of Concept Testing

Section 7 – Conclusion, lessons learned and next steps for FSA.

Appendix A – Identity and Access Management Prototype VDC Installation Issues

Appendix B – Tivoli Access and Identity Manager Hardware Requirements

Appendix C – eZAudit Logon Flow

Appendix D & E – Tivoli Access and Identity Manager Test Plan Outline

Appendix F & G – Tivoli Access and Identity Manager Test Results

Appendix H – Code Examples for eZAudit Integration

Appendix I – TIM & TIM HP UX11i Hardware Specs for IBM Test Lab Installation

3 Identity and Access Management Solution

3.1 Solution Vision

Figure 1 depicts the Enrollment and Access Management solution vision for FSA that was developed in November 2003 as a part of the Data Strategy Task Order. This solution vision illustrates the key components of the solution and the business benefits associated with each component. (The Data Strategy project defined this overall capability as 'Access Management', to include Identity Management and Access Control functions. In the current Task Order, the terminology was changed to 'Identity and Access Management', to be more consistent with terminology in common use by vendors and the security industry.)

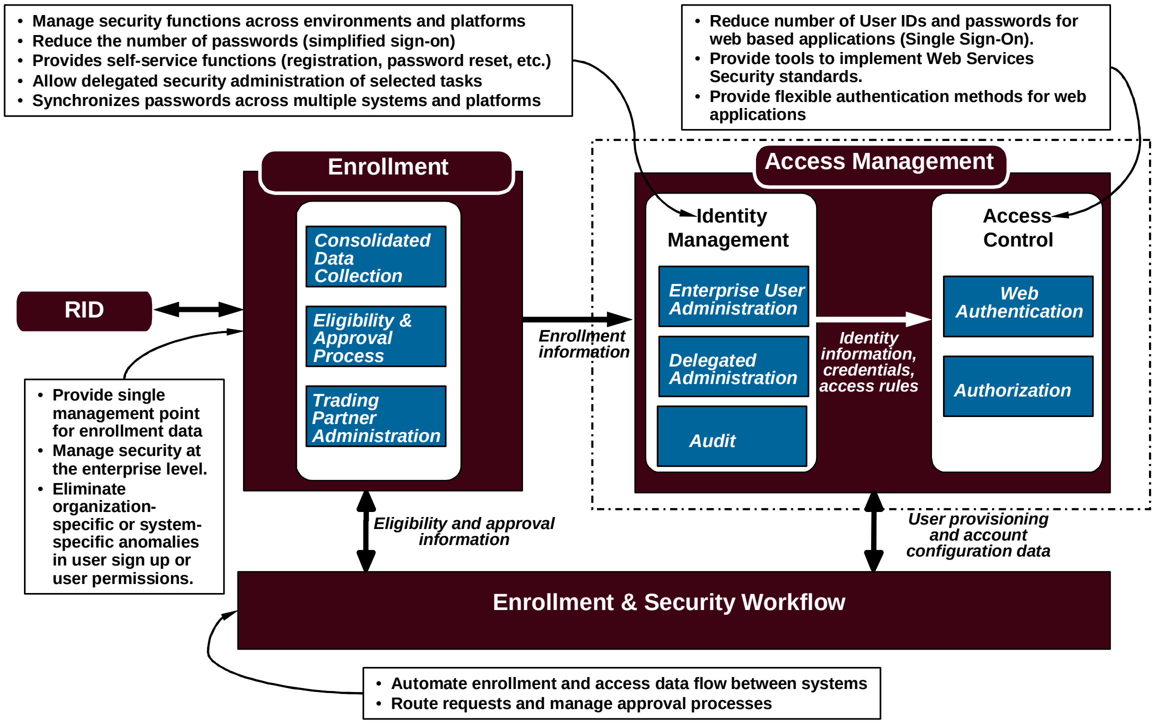

• Manage security functions across environments and platforms
• Reduce the number of passwords (simplified sign-on)
• Provides self-service functions (registration, password reset, etc.)
• Allow delegated security administration of selected tasks
• Synchronizes passwords across multiple systems and platforms
• Reduce number of User IDs and passwords for web based applications (Single Sign-On).
• Provide tools to implement Web Services Security standards.
• Provide flexible authentication methods for web applications
Enrollment
Consolidated Data Collection
Eligibility & Approval Process
Trading Partner Administration
RID
Enrollment information
Access Management
Identity Management
Enterprise User Administration
Delegated Administration
Audit
Identity information, credentials, access rules
Access Control
Web Authentication
Authorization
• Provide single management point for enrollment data
• Manage security at the enterprise level.
• Eliminate organization-specific or system-specific anomalies in user sign up or user permissions.
Eligibility and approval information
User provisioning and account configuration data
Enrollment & Security Workflow
• Automate enrollment and access data flow between systems
• Route requests and manage approval processes

Figure 1 – FSA Enrollment and Access Management Solution Vision

3.2 Prototype Tool Solution

FSA selected the IBM Tivoli Access Manager 5.1 and IBM Tivoli Identity Manager 4.5.1 products for the Web Access Control and Identity Management Proof of Concept. These security tools provide the functionality necessary to meet FSA's business objectives for Identity and Access Management.

3.2.1 Tivoli Access Manager Version 5. 1

General Description

IBM Tivoli Access Manager provides Web single sign-on, distributed Web-based administration, and policy-based security. Tivoli Access Manager includes the following components:

- Access Manager Policy Server maintains the master authorization database for the secure domain. This server processes access control, authentication, and authorization requests.
- Authorization Database (Proprietary database bundled with the Policy Server) is used for authorization functions. It is separate from the User Registry and contains a virtual representation of the resources it protects.
- WebSEAL Server is a reverse proxy that applies a security policy to a protected resource. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy.

Business Benefits

Tivoli Access Manager provides multiple benefits for both the trading partner user and FSA. This technology will:

- Reduce the number of User IDs and passwords and enable Single Sign-On functions for web-based applications.
- Provide tools to implement Web Services Security standards.
- Provide flexible authentication methods for web applications.
- Allow security functions (e.g., authentication, authorization, auditing) to be implemented as consolidated infrastructure services instead of duplicated functions within each web application.
- Decrease the time required for design and implementation of new web applications by greatly reducing development time associated with security functions.
- Provide a single database for storing security data (user information, authentication information, access rules, etc.) to simplify security administration and maintenance for web applications.
- Provides a single point of entry into the application environment
- Increase system flexibility by providing access to complex security functions when needed without requiring custom development (e.g., strong authentication, digital certificate functions, e-Authentication compatibility) and allowing for varying levels of authentication within the application landscape

3.2.2 Tivoli Identity Manager Version 4.5.1

General Description

Tivoli Identity Manager (TIM) interacts directly with users and with two external types of systems: identity sources and access control mechanisms. The identity systems deliver authoritative information about the users that need accounts. The provisioning system communicates directly with access control systems to create accounts, supply user information and passwords, and define the entitlements of the account. In reverse, local administrative changes made to an access control system are captured and reported to the provisioning system for evaluation against policy. Other features include:

- Role-based delegated administration allows administrative privileges to be distributed over organizational and geographical boundaries.
- Centralized Web administration
- Self-service interfaces remove the need for administrative personnel for password resets, password synchronization and the modification of personal information.
- Embedded provisioning engine and universal integration tools automate administrative tasks.
- A limited license for IBM Directory Integrator is bundled with TIM to allow for integration of disparate data sources without significant coding of adapters or custom agents

Business Benefits

Tivoli Identity Manager primarily benefits system administrators and system owners in the form of time and cost savings for operations such as password resets and improved audit capabilities. Some of the major benefits include the ability to:

- Integrate with and manage security functions across environments and platforms to enable development of enterprise user access roles.
- Improve the accuracy of assigning and monitoring access privileges across multiple systems.
- Reduce the number of user passwords (simplified sign-on) through automating synchronization of passwords across multiple systems.
- Deploy self-service functions such as automated password reset and user updates of demographic information.
- Allow delegation of selected security administration tasks to authorized remote administrators.
- Enhance enterprise auditing and reporting capabilities through creation of cross-system access reporting.
- Application and platform specific agents allow for rapid development and configuration for provisioning of standard platforms and custom applications

4 Prototype Design and Implementation

4.1 Prototype Overview

The plan for this effort was to install Proof of Concept instances of Tivoli Identity Manager, Tivoli Access Manager and the WebSEAL proxy to simulate deployment of security tools during an FSA enterprise deployment. Team members, acting as system security administrators, would set up users in TIM and grant them access to the eZAudit and Experimental Sites target applications. These authorizations would then be sent to the TAM Policy Server and enforced by WebSEAL proxy server. To integrate applications with TAM, authentication functionality would be removed from each application and provided by the WebSEAL Proxy and Tivoli Access Manager components. Use of WebSEAL Proxy and Tivoli Access Manager would also enable Single Sign-on between the target applications.

The following diagram is a logical design for the planned prototype components including the TIM Administrative Interface, TAM WebSEAL Proxy Server, TAM Policy Server and a sample target application (in this case eZAudit).

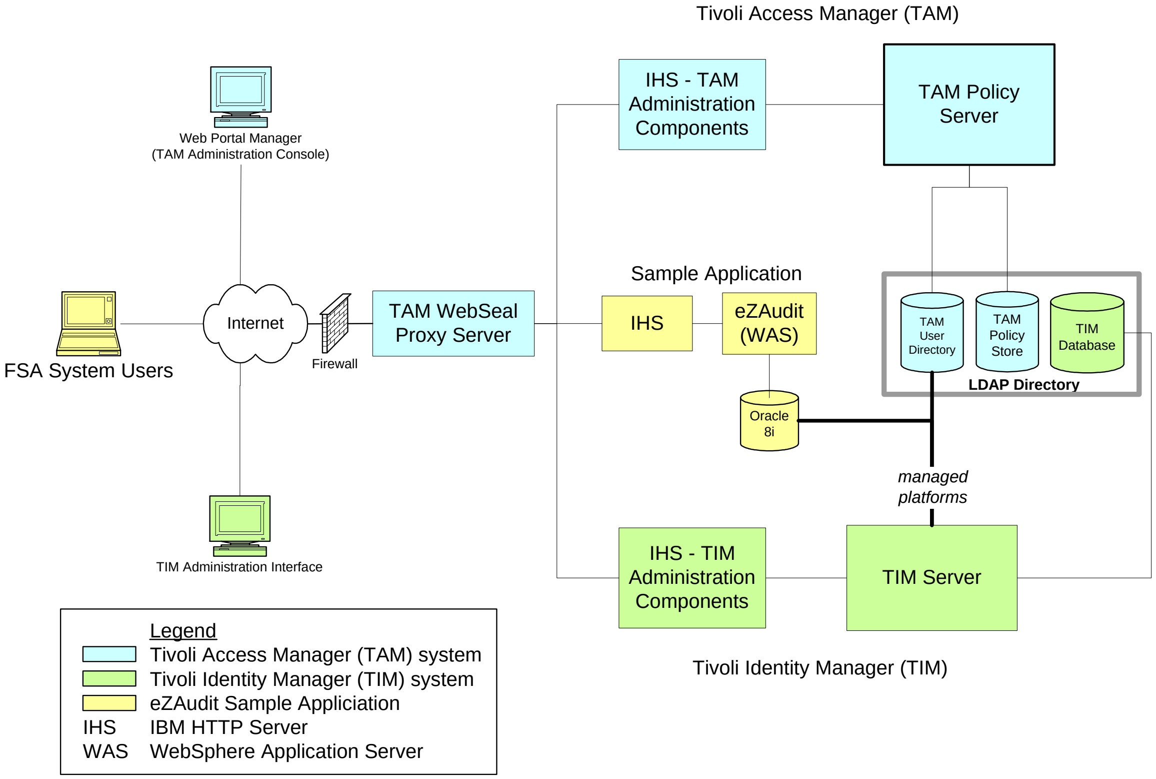

Tivoli Access Manager (TAM)
Web Portal Manager
(TAM Administration Console)
IHS - TAM Administration Components
TAM Policy Server
Sample Application
FSA System Users
Internet
Firewall
TAM WebSeal Proxy Server
IHS
eZAudit (WAS)
Oracle 8i
TAM User Directory
TAM Policy Store
TIM Database
LDAP Directory
managed platforms
TIM Administration Interface
IHS - TIM Administration Components
TIM Server
Tivoli Identity Manager (TIM)
Legend
Tivoli Access Manager (TAM) system
Tivoli Identity Manager (TIM) system
eZAudit Sample Appliciation
IHS IBM HTTP Server
WAS WebSphere Application Server

Figure 2 – Logical Prototype Diagram

For additional information on installing the TIM and TAM products, the installation reference guides for TIM and TAM are available at the following web sites:

TIM: http://publib.boulder.ibm.com/tividd/td/IdentityManager4.4.html

TAM: http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business5.1.html

TAM Integration with TIM: http://publib.boulder.ibm.com/tividd/td/ITAME/SC32-1364-00/en_US/HTML/am51_tim_guide02.htm

4.2 Prototype Schedule

In order to ensure a successful security tools Proof of Concept, dependencies and lead times for coordination were identified in the project plan. The initial schedule was:

Date	Task
3/05/04	FSA finalizes security tool selection
3/08/04 -4/01/04	Finalize design of Proof of Concept, submit requests to VDC for hardware, VDC prepares hardware, make arrangements for Proof of Concept installation.
4/01/04 - 4/16/04	Servers available for use, complete test of access/connectivity
4/01/04 - 4/16/04	Integration of Single Sign-on into one or two sample FSA Java based web apps and preliminary testing of application
4/19/04 - 4/23/04	Installation of TAM / TIM
4/26/04 - 5/04/04	Configuration of TAM / TIM and integration with one or two sample FSA Java based web apps and preliminary testing
5/05/04 - 5/12/04	Test TAM and TIM in FSA environment.
5/10/04 - 5/14/04	Document Test Results.
5/14/04	Proof of Concept complete.

Figure 3 – Planned Prototype Schedule

Delays in final selection of the security products, hardware procurement, and coordination of permission to install the products with on-site vendor support, limited the time available for system integration and testing. The actual prototype schedule was as follows:

Date	Task
3/26/04	FSA finalizes security tool selection
3/29/04 -4/16/04	Finalize design of Proof of Concept, submit requests to VDC for hardware, VDC prepares hardware, make arrangements for Proof of Concept installation.
4/12/04 - 4/30/04	Integration of Single Sign-on into one or two sample FSA Java based web apps and preliminary testing of application
4/26/04 - 4/30/04	Servers available for use, complete test of access/connectivity
5/03/04 - 5/07/04	Installation of TAM / TIM at VDC
4/26/04 - 5/13/04	Configuration of TAM / TIM and integration with one or two sample FSA Java based web apps and preliminary testing
5/10/04 -5/13/04	Test TAM and TIM in FSA environment.
5/10/04 - 5/14/04	Document Test Results.
5/14/04	Proof of Concept complete.

Figure 4 –Actual Prototype Schedule

4.3 VDC Environment Setup

This section documents the proposed design utilizing the ITA development environment and the actual installation steps completed. FSA requested that the prototype be installed in the FSA Integrated Technical Architecture (ITA) development environment located at the Virtual Data Center (VDC). Although the design is complete, technical difficulties with TIM and TAM at the VDC prevented the design from being successfully implemented.

As a result of security concerns, the VDC proposed an off-site installation using the SUDO utility to manage root level access. Due to the number of components and complexity of configuration steps required during these installations, IBM requested on-site root access. FSA approved IBM's request, the VDC granted on-site access with system and middleware administrator support.

The VDC made two HP-UX 11i servers available for the proof of concept and configured them according to the vendor's specifications. The basic server specifications for these two HP-UX 11i SuperDome N Class Servers:

- 8 x 360 CPUs
- 8 GB of Memory
- 2 / 9GB internal disks with external 175GB disk array

The team intended to install each of the components on individual servers in order to more closely match the Proof of Concept to a production environment. However, due to a lack of availability of a third HP-UX 11i machine, HPN1 was partitioned into HPN1 TIM and HPN1 TAM. HPN2 was used to host the web seal reverse proxy. The RDBMS agent would be installed on HPN1 TIM and the TAM agent would be installed on HPN1 TAM. Both of these servers were created from an existing WebSphere/MQ server image by CSC. Also, both servers were configured to the standard CSC/VDC development environment specifications to comply with audit/security standards.

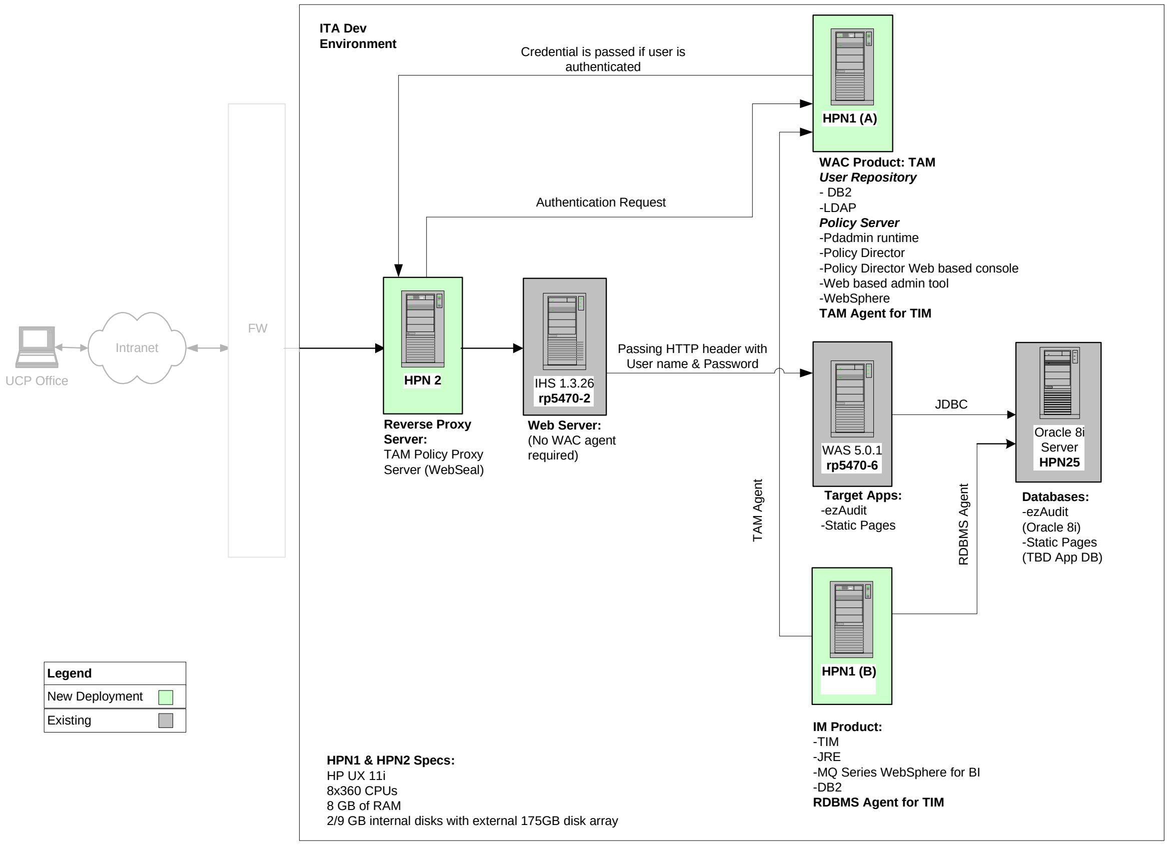


Figure 5 – Prototype VDC Environment Setup

It is important to note that this prototype design differs from an actual production deployment in several ways:

- TIM, TAM, and the WebSEAL Proxy Server would normally be installed on separate pieces of hardware and tuned for performance in a production environment.
- Individual LDAP/Directory Server instances would be installed on each WebSEAL instance and configured to be read-only replicas of the master instance.
- The proxy server and the web server would be located in a DMZ and separated from the internal network by a firewall in a production environment.
- SSL would be enabled for encrypted communications between components in a production environment.

The applications selected for modification, eZAudit and Experimental Sites, were redeployed on the existing application/database servers on rp5470-6 and HPN25 respectively. Given the shared nature of the WebSphere and Oracle servers, the goal of the team was to make as few changes as possible to existing servers. The prototype team obtained the source of each application for use in the Proof of Concept. The modifications made to the target applications are described in detail in Section 5 – Application Integration.

4.4 Installation at the VDC

The attempted installation of Tivoli Access Manager and Tivoli Identity Manager occurred at the VDC in Meriden, Connecticut from May 3 – May 7, 2004 and was performed by IBM. The following individuals participated in the installation effort.

Name	Role	Email Address	Onsite
Srinivas Kankanahalli	FSA Business Lead	Srinivas.Kankanahalli@ed.gov	Yes (5/4-5/6)
Ryan A. Summers	Accenture Project Lead	Ryan.a.summers@accenture.com	No
Anthony Ko	Accenture Technical Lead	Anthony.c.ko@accenture.com	Yes (5/4-5/6)
Paul Izzo	Unix/System Administrator	pizzo3@csc.com	Yes
Michael McCarey	Middleware Administrator	mmccarey@csc.com	Yes
Laura Mueller	CSC Project Manager	lmueller@csc.com	No
Stephen M Byrnes	Security Software Sales Specialist	sbyrnes@us.ibm.com	Yes (5/4)
Joseph E. Hamblin	Technical Security Specialist	jhamblin@us.ibm.com	No, On Standby
William Scott Downs	Technical Security Specialist (TAM)	wsdowns@us.ibm.com	Yes (5/3-5/7)
Robert B. Adachi	Technical Security Specialist (TIM)	radachi@us.ibm.com	Yes (5/4-5/7))

Figure 6 – TAM/TIM Installation Team Members

As depicted in the VDC server landscape, the installations of TIM and TAM require the installation of TIM and TAM specific versions of DB/2, LDAP, MQ and WebSphere. IBM bundles these middleware components as part of the TIM and TAM license and does not recommend the installation of different versions of the components. The supplied middleware components are customized specifically for TIM and TAM.

In order to comply with VDC server security and audit rules, the HP-UX 11i servers were created with the standard WebSphere/DB2 server images. IBM stressed that all vestiges of the WebSphere/DB2 installation must be removed to avoid any potential library/configuration conflicts. Although WebSphere and DB2 were then removed from the servers by a system administrator, there were some pieces of the software that had to be removed manually by IBM.

The initial installation of TIM/TAM was expected to take three days, based on IBM's estimate. TAM was scheduled to take one day and TIM would take two days. The actual Tivoli Access Manager installation attempt occurred from May 3-7, 2004. The Tivoli Identity Manager attempt occurred from May 4-7, 2004. However, issues related to TIM and TAM running on HP-UX 11i could not be resolved completely during the week allocated for on-site access.

The main issues during installation were related to the fact that HP-UX 11i is not a commonly supported platform for IBM. As release 5.1 of Tivoli Access Manager and release 4.51 of Tivoli Identity Manager were the first versions to support HP-UX 11i, there was concern that the issues were related to the installation of more current HP-UX 11i patches at the VDC than what was certified by IBM. Policy Director (a component of TAM) could not be installed in the VDC environment due to an inability to read SSL certificates. TIM DB/2 would not start up under HP-UX11i after all the required patches were applied. A complete list of results and issues encountered during the five day VDC on-site installation is included in this document as Appendix A – Identity and Access Management Prototype VDC Installation Issues.

4.5 Installation at UCP

Given the installation issues at the VDC, the prototype team requested installations of the Windows versions of TIM and TAM to deploy locally at the Union Center Plaza FSA location. The goal was to ensure a working Proof of Concept installation of both TIM and TAM so that they would be available for integration testing. IBM provided a computer with TAM and WebSEAL pre-installed and a VMWare image of a server with TIM installed.

Although IBM provided fully functional installations of TIM and TAM, they were not already integrated. In addition the required TIM agents were not initially provided by the vendor, although they were later supplied upon request.

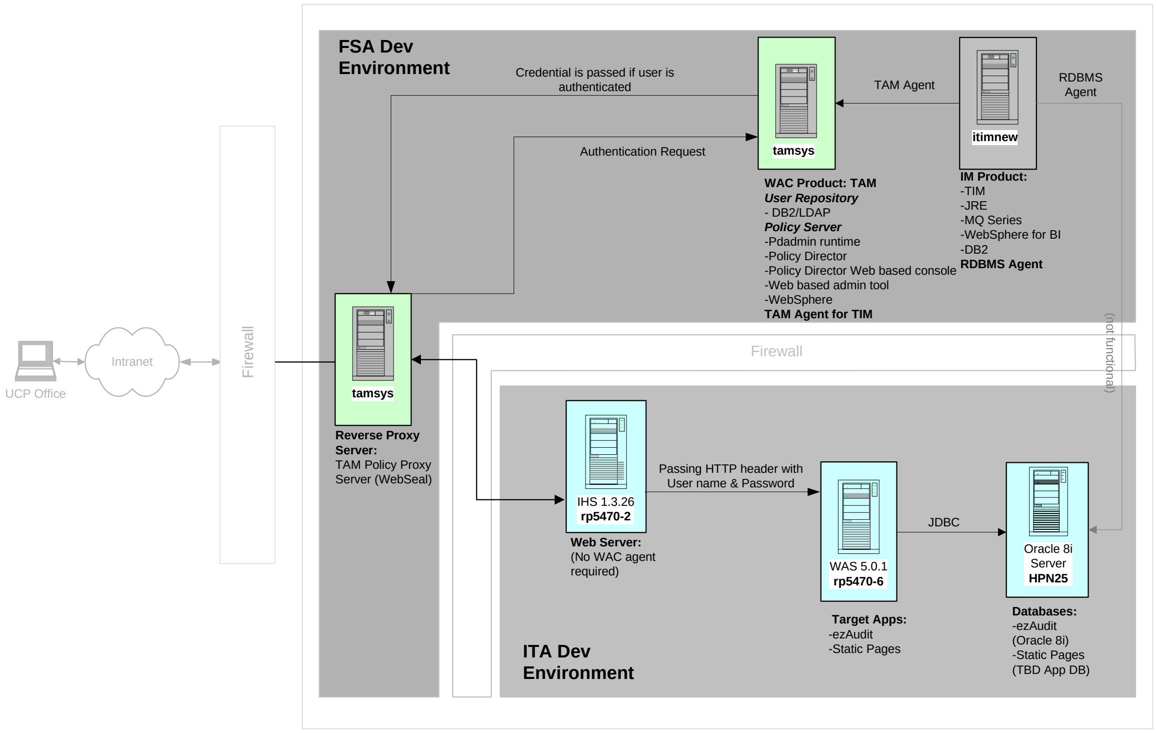


Figure 7 – Prototype FSA/VDC Environment Setup

In this implementation, TIM and TAM were set up in the FSA environment to control access to the eZAudit and Experimental Sites web applications hosted in the VDC. The functional architecture was not affected by the environment changes because the servers in the VDC network do not initiate communication with servers in the FSA network.

4.6 Installation at IBM Labs

TIM 4.5.1 and TAM 5.1 were the first versions that officially supported HP-UX 11i. After encountering multiple issues during the installation of TIM and TAM on the HP-UX 11i servers at the VDC, IBM investigated the installation difficulties in their development labs in Irvine, California and Raleigh, North Carolina. Developers and quality assurance personnel from both the TIM and TAM product teams were involved.

After analyzing the TIM and TAM server logs from the VDC servers and replicating the server configurations, IBM was able to successfully install TIM and TAM on HP-UX 11i in their test labs. However, based on the information provided by IBM, the HP-UX 11i patches used in the IBM test lab can not be directly compared to those installed on the servers at the VDC. IBM used the most current service packs as of May 2004 and the servers at the VDC used service pack from December 2002 in conjunction with additional patches. The IBM test servers also do not include the additional monitoring and maintenance software present on the VDC servers. Further testing should be conducted prior to future installation attempts in the ITA environment.

Additional details of this test lab install are provided in Appendix I – TAM & TIM HP-UX 11i Hardware Specs for IBM Test Lab Installation.

4.7 Tivoli Access Manager Configuration

4.7.1 User Flow

This Prototype User flow steps through the log in process:

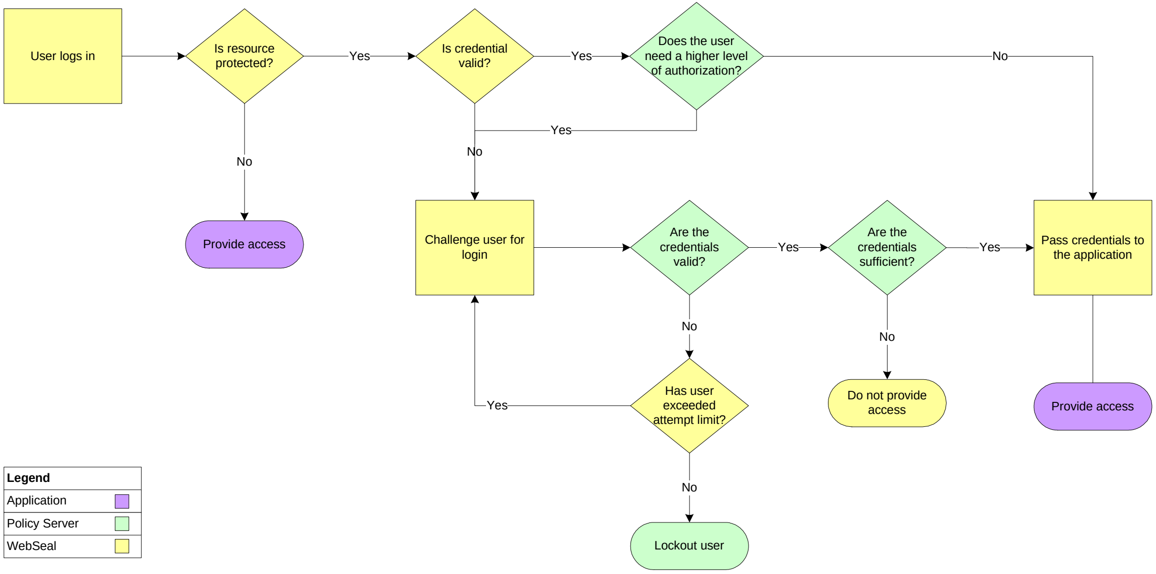


Figure 8 – Prototype User Flow

When a user attempts to access a protected application, the request would be intercepted by the TAM Proxy Server. If the user had not logged into the single-sign on environment, the proxy would challenge the user for their user name and password. A user would be allowed three attempts to log in.

If successful, the proxy server sends the credentials to the protected application and tracks the user's credentials. At this point, the user would be able to access any protected application in the single-sign on environment that they have authorization to access. If they did not have authorization to access the application, an error page is displayed stating they do not have sufficient access.

4.7.2 WebSEAL Proxy

By acting as a reverse proxy, all web traffic within the Single Sign-on environment is directed through a single server. *Junctions* then map a host name to a path on the web seal server.

In a non-WebSEAL environment, the user would log directly into the web application, using an http link such as:

```
http://ita.ezaudit.ed.gov:8533/EZWebApp/login.do
```

```
host name               path name
```

In a WebSEAL protected environment, the user would log in to the application through the proxy, using a link such as:

```
http://tamsys/eZAudit/EZWebApp/login.do

                junction name  path name
```

WebSEAL junctions for eZAudit and Expsite are created in WebSEAL and associated with the "eZAudit_user" and "Expsite_user" ACLs. This enforces an authorization policy on the entire web application defined in the junction.

To allow for finer grain placement of access control lists, TAM uses a CGI script called "query_contents" to determine the web and application space of the server. For the purpose of the Proof of Concept, we did not elect to have the CGI script installed because of the additional steps required to get software loaded on servers located at the VDC. CGI scripts are not normally deployed in a WebSphere environment. As such, the entire web junction is treated as a single protected resource.

4.7.3 Access Control

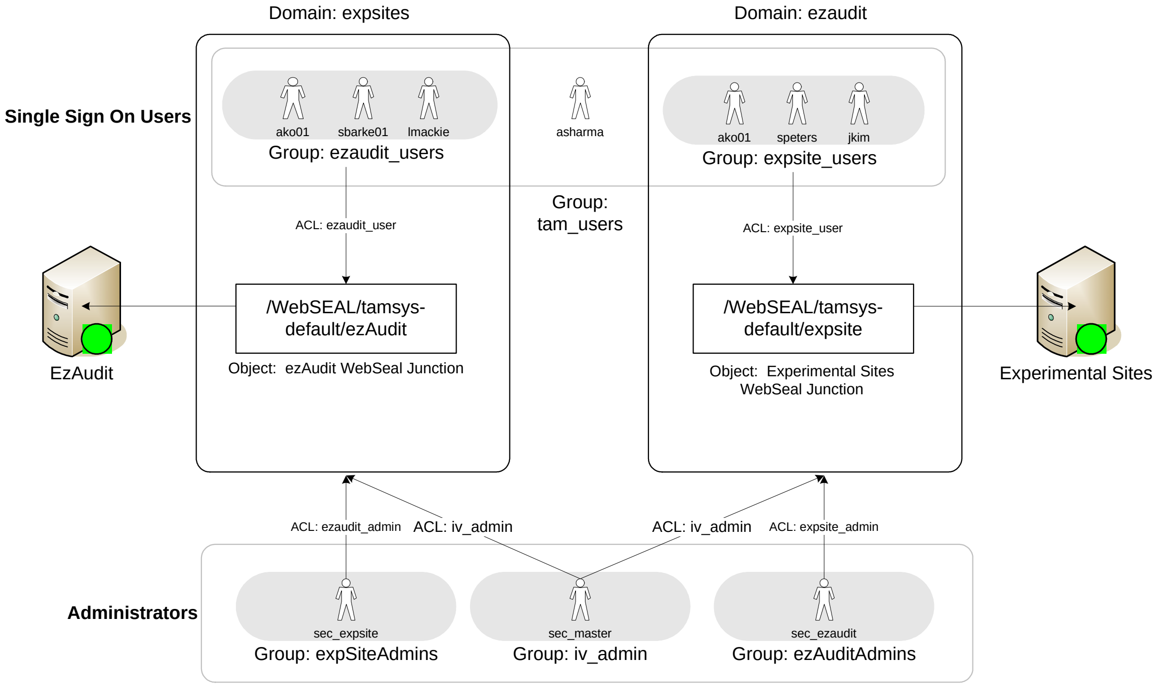


Figure 9 – Tivoli Access Manager Configuration

Figure 9 depicts the configuration used for setting up Tivoli Access Manager. Domains for "Experimental Sites" and "eZAudit" are created in TAM Web Portal Manager. Each domain has a group of administrators and users. For example, Single Sign-on to the application is controlled by the "eZAudit_user" and "Expsite_user" access control lists and administration of the user groups "ezaudit_user" and "Expsite_user" are respectively controlled by the "eZAudit_admin" and Expsite_admin" ACLs. All Single Sign-on users belong to the "tam_users" group. The central administration group is called "iv_admin."

In a Single Sign-on scenario user "ako01" can get access to both the eZAudit and the Expsite application, since that user is a member of both domains. User "sbake01" is only a member of the Experimental Sites domain and can only access that application. WebSEAL will not allow this user to access eZAudit. User "jkim" is only a member of the eZAudit domain and will not be able to access Expsite application. Note that the user "asharma" has not been placed in any group and therefore does not have access to either application but is authenticated to the Single Sign-on environment.

In an actual implementation, it is possible to use the Global Sign On functionality to send unique username and passwords to each system. A mapping is created within TAM that creates a sub-account for each user. This works independently of a provisioning solution.

4.8 Tivoli Identity Manager Configuration

4.8.1 Roles and Provisioning Policies

Within TIM, organizational roles define the job functions and policies define the properties that define the entitlements. Organizational roles can be static or dynamic. Static roles have to be directly assigned to a user. Dynamic roles are assigned based on criteria specified by the administrator. For example, a role can be defined such that anyone who has a specific title is assigned to it.

Each policy is independent and can be keyed to organizational roles and agent service targets. They can be set to run automatically or manually and the enforcement of the policy can be configured to be default, mandatory, optional or excluded. Default values only apply to new users for which a policy applies. A mandatory policy automatically corrects the user record to bring it in line with policy. Optional and excluded policies allow administrators to define what values can or cannot be selected for a given value. Workflows can also be associated with a policy for approval.

When a policy is created with mandatory enforcement, it runs as soon as it is enabled. Also, each policy has an associated priority and is executed in order based on how high the priority is. As a result, it is critical that policies are documented and planned such that they do not override other policies or are so broadly defined that they overwhelm the system.

4.8.2 Workflow

In the Proof of Concept, one centralized administrator is responsible for handling all access requests. In a production system, one can configure manual workflow processes that require one or more people to approve the request. One can set up multiple levels of workflow and also escalation policies if the requests are not handled in a timely manner. Furthermore, TIM allows the administrator to delegate their responsibilities to other administrators and build a federated security model. All administrators will have the ability to monitor their approval queues through the TIM front-end or receive email notifications when a work item is assigned to them.

4.8.3 Provisioning Agent Configuration

In addition to the base provisioning server, TIM requires agents that communicate with the TIM server to handle communication with integrated applications/platforms. In this Proof of Concept, the following agents are required:

Agent Name	Installation Location	Purpose
RDBMS Agent	TAM Server. (Must be installed on Windows 2000)	Enables provisioning and reconciliation with the Oracle databases for eZAudit
TAM Agent	TAM Server	Enables the provisioning of TAM Single Sign-on users and their policies.

The agents communicate with the TIM server via Directory Service Markup Language (DSML) over a SSL connection or via FTP. As all client/server communication is encrypted, each agent and the TIM server must be configured with security certificates created by the same Certificate Authority (CA) authority. If using DSML, a port must be assigned for TIM to communicate with the agent.

These agents can function in both a push and pull capacity. Reconciliation can be initiated through the TIM provisioning front-end or pushed to TIM via agent defined event notifications. As an example, the TAM agent can be configured to create a new TIM user if they are added directly via the TAM front-end. These users can be automatically assigned to a user group in TIM or marked as orphans for manual assignment.

It is important to note that agents are not available on all platforms supported by the Identity Manager server. As examples, the RDBMS and Oracle agents run only on Windows, the HP-UX agent runs only on HP-UX and the RACF server runs only on OS/390 or z/OS. If the deployment of all agents on HP-UX is required, IBM suggests that the Directory Integrator service be used. In fact, IBM has planned to move away from agents and standardize on Directory Integrator for provisioning because of its platform independence.

4.8.4 Password Management

A single password will be used for all integrated applications in the TIM/TAM environment. In the Proof of Concept design, we will specify the passwords in TIM and push them to the source systems during reconciliation. It is possible to maintain separate passwords for each source system through direct provisioning via TIM and the use of the

Global Sign On functionality in TAM. For the purposes of this Proof of Concept, this functionality will not be necessary.

It should be noted that IBM does not recommend the use of separate passwords in a Single Sign-on environment where users can maintain user information on the source application. The risk is that the passwords will become unsynchronized. Furthermore, it would not be possible to determine an existing password since it has been stored as a secure hash. In a scenario where only TIM is used but not TAM it could be possible to push the password change back to the managed platform. With this implementation approach, the password change would be handled by the RDBMS agent.

4.9 Tivoli Access Manager / Identity Manager Setup and Migration

Both Tivoli products store configuration and identity data as entries in separate LDAP instances. The initial installation process creates custom LDAP classes and suffixes that are uniquely identified with an object identifier (OID). It is important to note that the configuration of TIM and TAM are tied to the specific machines on which they are installed. This has implications for planning the migration of configurations from server to server and from environment to environment. This would also prevent direct migration of the Windows based TIM and TAM instances to a HP-UX platform.

Within TAM, configuration values are generated within the master Policy Server. When an object is created in TAM, they are assigned a unique identifier called “secUUID.” This means that all user, group and policy information are tied to an arbitrary value. As a result, it is not possible for one to extract policy and group information and migrate it to another server. The recommendation by IBM is to record the commands issued to the PDAdmin command line to build a script but this is obviously not an ideal solution.

TIM also generates UUID for some objects called “erglobalid”. However, it provides somewhat more robust tools for exporting and importing information from the LDAP server. However, these tools are little more than tools to generate LDIF files. The execution of the policies and provisioning would have to be done at each stage of deployment.

It is critical that the development and deployment of a Tivoli I& AM solution must be well planned and documented. The dynamic nature of provisioning and policy enforcement will require rigorous regression testing to ensure that unexpected errors in security do not happen.

5 Application Integration

5.1 TAM / Application Authorization Integration

The Proof of Concept focuses on integrating Single Sign-on and centralized user management into existing applications. In the current environment, authorization and authentication functionality is handled by each application. In the figure below, each attempt to log on results in a query to an application specific database to verify a given user name and password.

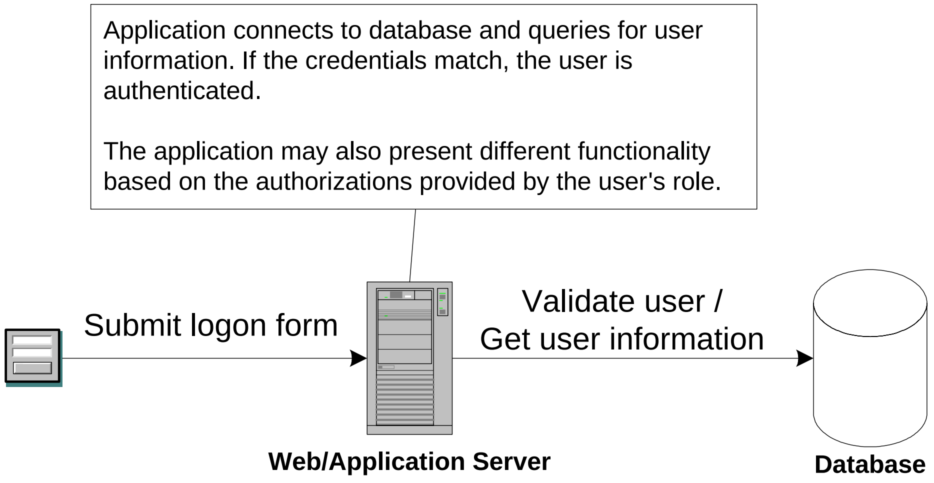
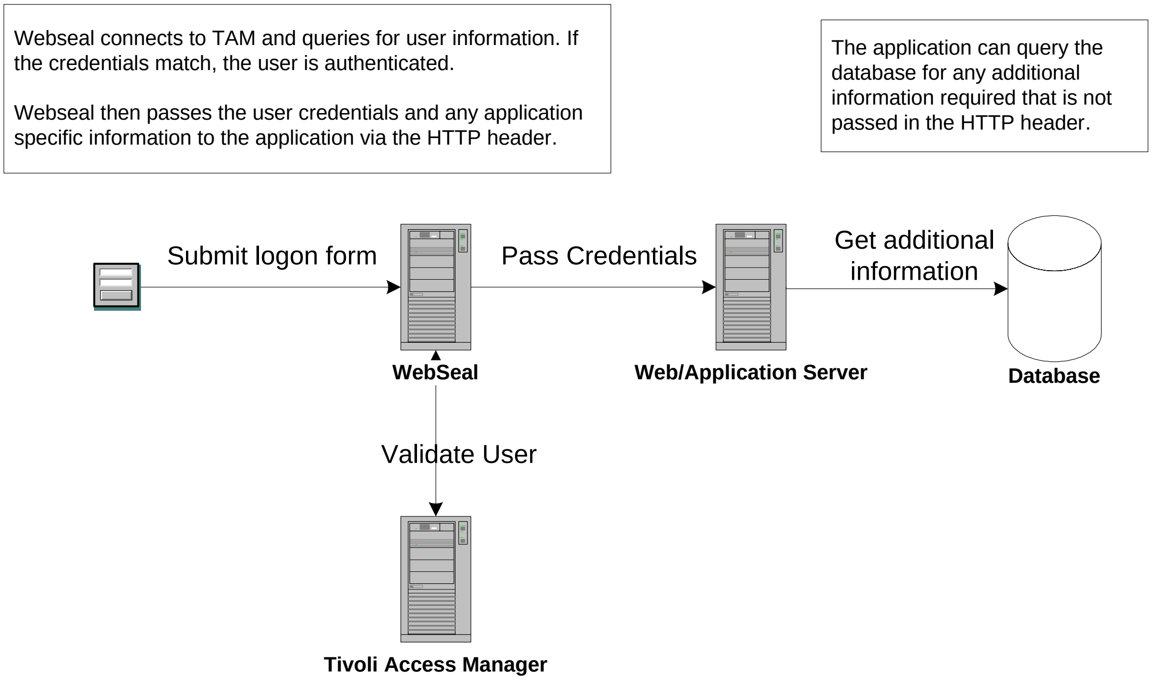

Application connects to database and queries for user information. If the credentials match, the user is authenticated.
The application may also present different functionality based on the authorizations provided by the user's role.
Submit logon form
Validate user / Get user information
Web/Application Server
Database

Figure 10 – Standard Application Logon Flow

In the Single Sign-on environment, the logic required to handle authentication is moved from the application to the access control product.



Webseal connects to TAM and queries for user information. If the credentials match, the user is authenticated.

Webseal then passes the user credentials and any application specific information to the application via the HTTP header.

The application can query the database for any additional information required that is not passed in the HTTP header.

Figure 11 – Single Sign-on Application Logon Flow

There are three possible ways for WebSEAL to send the authorization information to the application server.

1) Pass user credentials through the HTTP Header.
2) Pass the credentials programmatically through JAAS or via J2EE container-level authorization.
3) Pass the credentials via Lightweight Third-Party Authentication (LTPA) (this option is specific to the WebSphere platform)

We have selected the first method because it is a generic approach that works for most web applications and do not require the inclusion of any additional libraries.

When the credentials are passed by WebSEAL to the application over the HTTP header, additional information can also be sent provided it fits within the size of the header. This can be in the form of additional header name/value pairs or encoded into the Extended Privilege Attribute Certificate (EPAC). This will result in increased performance because the application does not need to make additional queries to the database related to the authentication and authorization process.

5.2 TIM / Application Provisioning Integration

The second change to the application involves modifying how user administration is handled. Without centralized user administration, each application has its own user store as depicted below.

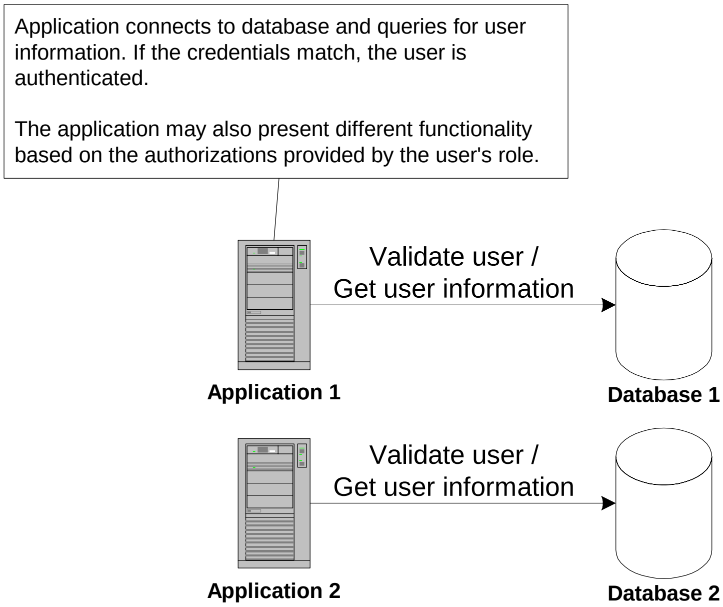

Application connects to database and queries for user information. If the credentials match, the user is authenticated.
The application may also present different functionality based on the authorizations provided by the user's role.
Validate user / Get user information
Application 1
Database 1
Validate user / Get user information
Application 2
Database 2

Figure 12 – Individual Application User Stores

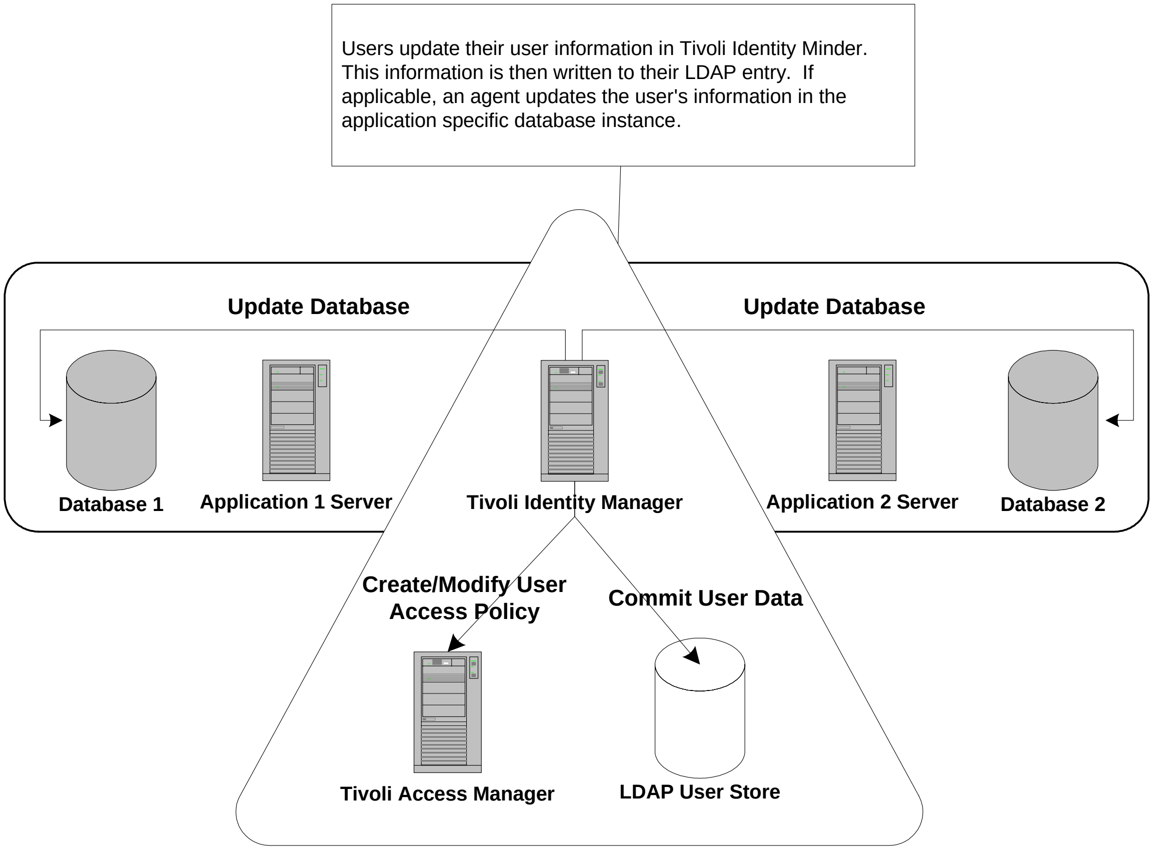
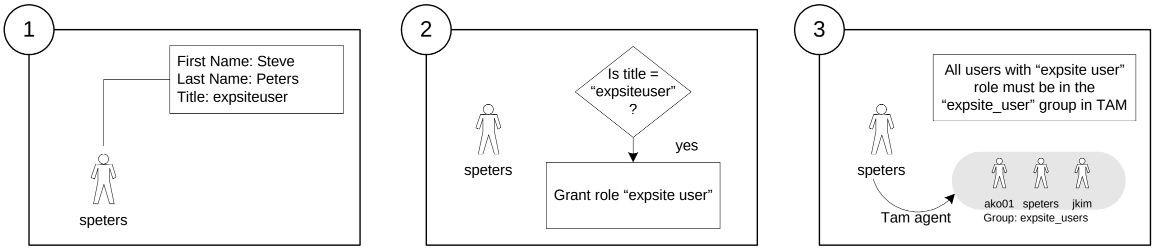

Users update their user information in Tivoli Identity Minder. This information is then written to their LDAP entry. If applicable, an agent updates the user's information in the application specific database instance.
Update Database
Update Database
Database 1
Application 1 Server
Tivoli Identity Manager
Application 2 Server
Database 2
Create/Modify User Access Policy
Commit User Data
Tivoli Access Manager
LDAP User Store

Figure 13 – User Information Creation/Updates in TIM

As part of TIM, there is a central LDAP directory that contains information for all users within the provisioning environment. This is done by the creation of a "person" entity within LDAP. TIM can then be setup to store any application/platform specific information by the creation of an "account" entity that shares common data, usually the "eruid", with the person identity. Each account can have application / platform specific data that is mapped fields to standard LDAP fields or stored in a custom LDAP class for the application.

For each service that TIM is provisioning, users can be migrated to TIM through reconciliation via the agent or loaded through IBM Directory Integrator (IDI). As mentioned earlier, reconciled users can be automatically grouped or created as orphaned users in TIM. However, during an initial migration, the usage of IDI to handle an extract/transform/load of source system users to TIM is recommended because IDI allows more control over how users are created in TIM. The administrator can select which fields are used to determine the identity of a user. A service is built into TIM to process the user data from IDI.

5.2.1 TIM Provisioning Process


1
First Name: Steve
Last Name: Peters
Title: expsiteuser
speters
2
Is title = “expsiteuser” ?
yes
speters
Grant role “expsite user”
3
All users with “expsite user” role must be in the “expsite_user” group in TAM
speters
Tam agent
ako01 speters jkim
Group: expsite_users

User Creation

Dynamic Assignment of “Expsite User” Role

Enforcement of “Expsite Tam Policy” Policy

Figure 14 – Provisioning Process for Experimental Sites

The above figure illustrates how the provisioning of Experimental Sites is handled. The dynamic role “Expsite User” has been set up to apply to any user with “Expsite_user” as the title. The “Expsite_tam_service” policy applies to all people with the “Expsite User” role and is mandatory. Since a newly created user will be out of line with the policy, TIM automatically grants them membership in the “Expsite_users” group in TAM. In an actual implementation, the flow would be extended to create a new user in the Experimental Sites database via a service using the Experimental Sites RDBMS or IDI agent.

5.3 Application Integration via Servlet Filter

Since FSA has standardized on the WebSphere servlet platform, applications can leverage existing authorization code across the enterprise. The authorization functionality for this Proof of Concept is built as a servlet filter that can be redeployed for any application with minimal modifications required. The code for this filter is in Appendix H – Code Examples for eZAudit Integration.

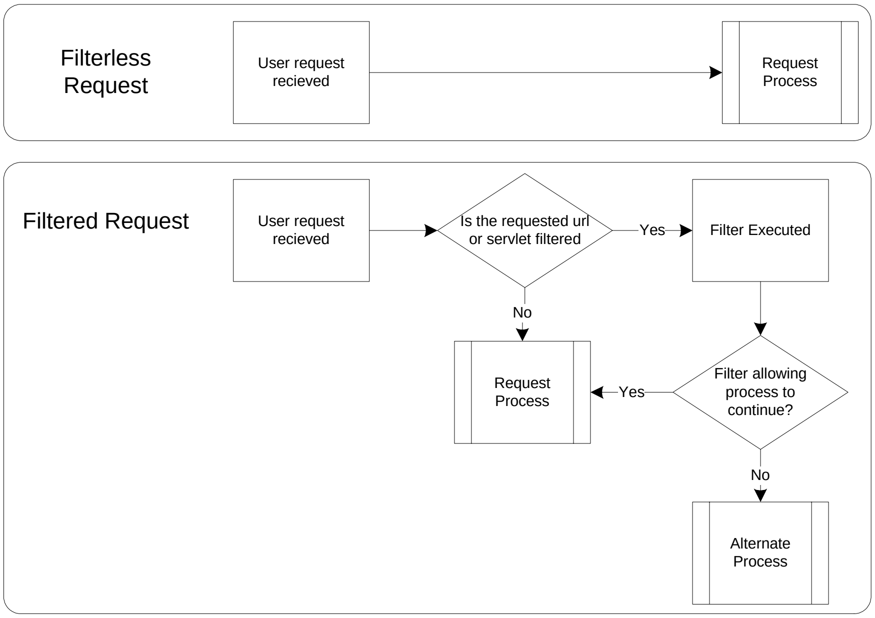


Figure 15 – Filtered vs. Filterless requests

There are three components to the authorization functionality:

- Authorization Servlet Filter – Used to intercept, validate and reject any unauthorized attempts to access the application. This is configured for specific URLs and servlets in the application's web.xml
- Application user context – Stores the user id, roles and credentials. Also stores a hash map of application specific values that are passed through the HTTP header
- Configuration file – Contains values that are application specific and do not impact the overall logic of the filter (i.e. url mappings for errors and any application specific header values that need to be placed into the user context hash map)

The filter dynamically intercepts HTTP requests and responses to transform or use the information contained in the requests or responses. Filters normally do not create responses, but instead provide universal functions that can be "attached" to any servlet or URL. They can intercept the request and forward the user to another response. This allows the authorization code to be easily ported to any application running on WebSphere. More importantly, it ensures that any functionality requiring authorization and authentication cannot run unless the filter is executed. This provides an additional level of security beyond what is offered by the WebSEAL proxy.

It is possible to enable basic Single Sign-on by simply configuring the filter to forward non-authenticated requests to the application log on page. Within the server environment, this also protects the application when a user attempts to log on directly to the application. Applications can then be deployed to the IBM WebSphere platform as EAR files without any further changes to the application server.

Filters also provide a convenient way to test web applications without requiring the Single Sign-on environment to be available. In the Proof of Concept, a test harness was constructed with an additional filter was created that would put dummy values into the HTTP request header.

5.4 eZAudit Integration

Application Background

eZAudit was developed to assist schools in submitting their required yearly audit information. eZAudit provides institutions with a way to submit financial statements and compliance audits through the internet. Designees from each school enter summary audit and financial data into a web form and attach an electronic version of the audit report for their school. The web-based submission allows for quicker processing by FSA and a faster response to the school.

Proof of Concept Integration with Tivoli Access Manager and Tivoli Identity Manager

Integration of single-sign on into eZAudit requires removing the existing authentication code and replacing it with the appropriate Tivoli functionality. Flow charts illustrating the old and new authentication and authorization processes are included as Appendix C – eZAudit Production and Prototype Logon Flows.

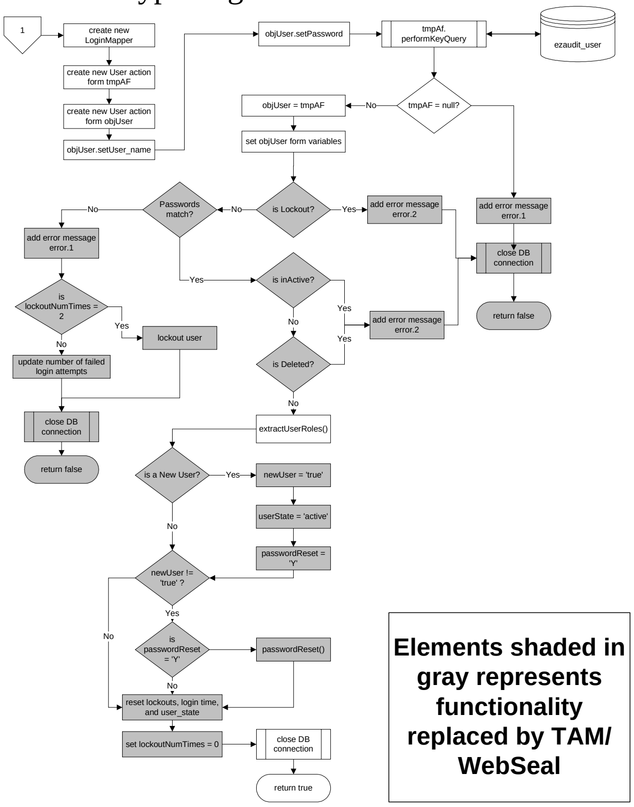


Figure 16 – Old Authentication Flow in eZAudit

In the old state, users are authenticated against data from an Oracle 8i database based on information entered via a web form. The user's input is tested against the hashed password in Oracle to determine if they match. If they do, additional logic runs to handle cases where their account has been disabled or locked out due inactivity. At that point, the application forwards the user to specific pages depending on what roles they are assigned.

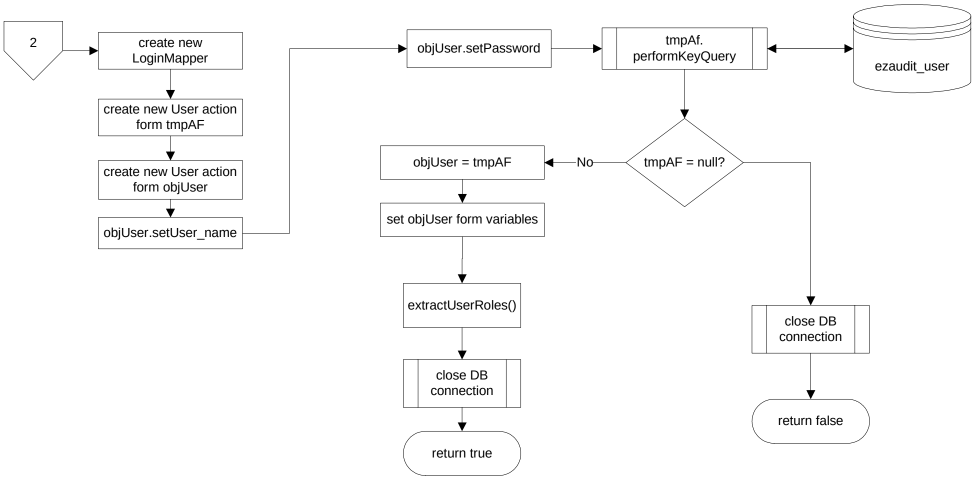


Figure 17 – New Authorization Flow in eZAudit

With the filter, the changes required to support Single Sign-on are localized to the login action class because the authorization process only occurs when the user accesses the login page. If authorized, a user context is created and the user can navigate to other pages.

In Proof of Concept, users are authenticated against the policy server via the WebSEAL proxy. If the user does not log in via WebSEAL, the user will be forwarded to a screen indicating they must log on through the WebSEAL proxy. User credentials are passed to the eZAudit application and are presumed to be valid. The user's role will be provisioned by TIM to the application database. Any login error, such as when a user has been set up in TAM but has not been provisioned in eZAudit, would result in an application exception. As a point for further integration, the roles a user has could be dictated by the groups they are assigned to in TAM.

5.5 Experimental Sites

Application Background

Experimental Sites is FSA's field-test for changes to specific Title IV statutory/regulatory requirements. 120 schools volunteered to participate in some of FSA's ten active experiments. Under the experiments, the schools are given exemptions to specific requirements governing student aid delivery in order to demonstrate how these exemptions can help schools streamline procedures/processes, improve student services, and eliminate delays in the delivery of Title IV aid. Participating schools are required to submit an annual report that captures performance-based data for the prior academic year (the report is due at the end of February). FSA then analyzes this data to evaluate the experiment results and their implications on how financial aid policy could be streamlined and simplified. In addition to program evaluation, this analysis is shared with policy makers in the Department of Education to assist in making informed decisions on recommendations for change in Title IV statutes and regulations.

Proof of Concept Integration with Tivoli Access Manager and Tivoli Identity Manager

The Experimental Sites application is a Java servlet application that is currently deployed on the Sun iPlanet platform. Work is currently underway to convert the application to a standard WAR that can run on IBM WebSphere. To reduce the time needed for integration into the Proof of Concept, Experimental Sites is only being used in a static capacity with URL based authorization. Any static content will be unprotected and any dynamic content will be protected. This will be accomplished by implementing multiple WebSEAL junctions and applying a servlet filter to specific URLs.

In this situation, the changes required to implement Single Sign-on are minor. As with the eZAudit integration, an authorization filter will be deployed to intercept the user information in the HTTP header. If the user is entitled to use Experimental Sites, they will be forwarded to a page indicating that Single Sign-on is successful. As with eZAudit, the user is not entitled or an error occurs during the log on process, the user will be forwarded to a screen indicating they must log on through the WebSEAL proxy.

6 Testing

6.1 Test Objectives

The testing plan for Tivoli Access Manager and Tivoli Identity Manager testing was designed to evaluate basic functionality to meet FSA requirements. The test plan outline for TAM is provided in Appendix D – Tivoli Access Manager Test Plan Outline. Tests conducted for Tivoli Access Manager included the following areas:

- Product installation
- Functionality testing
    - User authentication
    - Audit
    - Security policies

The test plan outline for TIM is provided in Appendix E – Tivoli Identity Manager Test Plan Outline. The testing for Tivoli Identity Manager included the following areas:

- Product installation
- Functionality testing
    - Creating new user accounts
    - Modify user accounts
    - Delete accounts or access
    - Audit
    - Security policies
    - Workflow

Some of the product functionality (e.g.; modify accounts/delete accounts) is common to both TIM and TAM, but these features will be tested in TIM to mirror the production environment as closely as possible. The scenarios and flows listed here will be tested as part of this Proof of Concept.

Application Authentication Testing

One of the main test objectives is to make sure that the application is being protected by Tivoli Access Manager. When a user tries to access a protected page, that request is directed to WebSEAL. Tivoli Access Manager checks the user credentials and grants access, if those credentials match.

The goal of the testing will be to make sure that the application is being protected by Tivoli Access Manager and that the user credentials are being checked. Figure 18 outlines this planned testing flow.

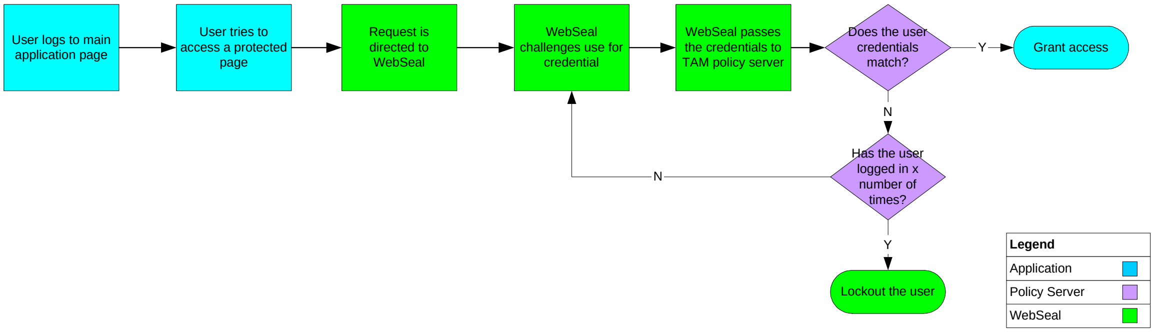

User logs to main application page
User tries to access a protected page
Request is directed to WebSeal
WebSeal challenges use for credential
WebSeal passes the credentials to TAM policy server
Does the user credentials match?
Y
Grant access
N
Has the user logged in x number of times?
N
Y
Lockout the user
Legend
Application
Policy Server
WebSeal

Figure 18 – Application Authentication Testing Flow

Single Sign-on Testing

Single Sign-on functionality allows a user to log into multiple applications without being re-challenged for credentials. In this testing scenario, user logs into application 1 and then tries to access a protected resource on application 2. User is granted access to application 2 without being challenged for credentials again if user has access to that resource. If the user does not have access to the protected resource on application 2, he is taken to the registration page. This testing flow is outline in figure 19

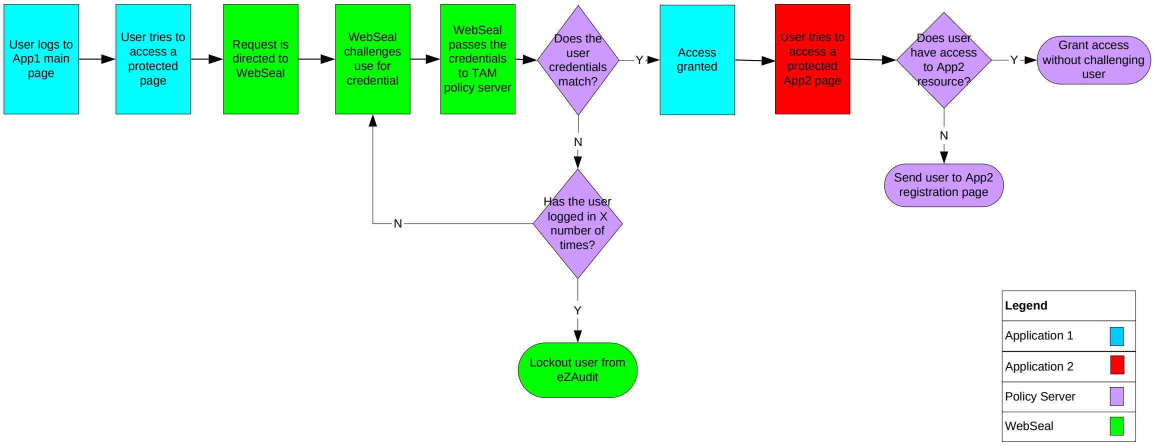

User logs to App1 main page
User tries to access a protected page
Request is directed to WebSeal
WebSeal challenges use for credential
WebSeal passes the credentials to TAM policy server
Does the user credentials match?
Y
Access granted
User tries to access a protected App2 page
Does user have access to App2 resource?
Y
Grant access without challenging user
N
Send user to App2 registration page
N
Has the user logged in X number of times?
N
Y
Lockout user from eZAudit
Legend
Application 1
Application 2
Policy Server
WebSeal

Figure 19 – Single Sign-on Testing Flow

Centralized Administration Testing

Test 1: Adding user to application 1

This scenario tests the administrative functionality of TIM. A centralized administrator adds a user to TIM. The test includes checking that the user can actually login and get access through TAM. Figure 20 depicts this flow.

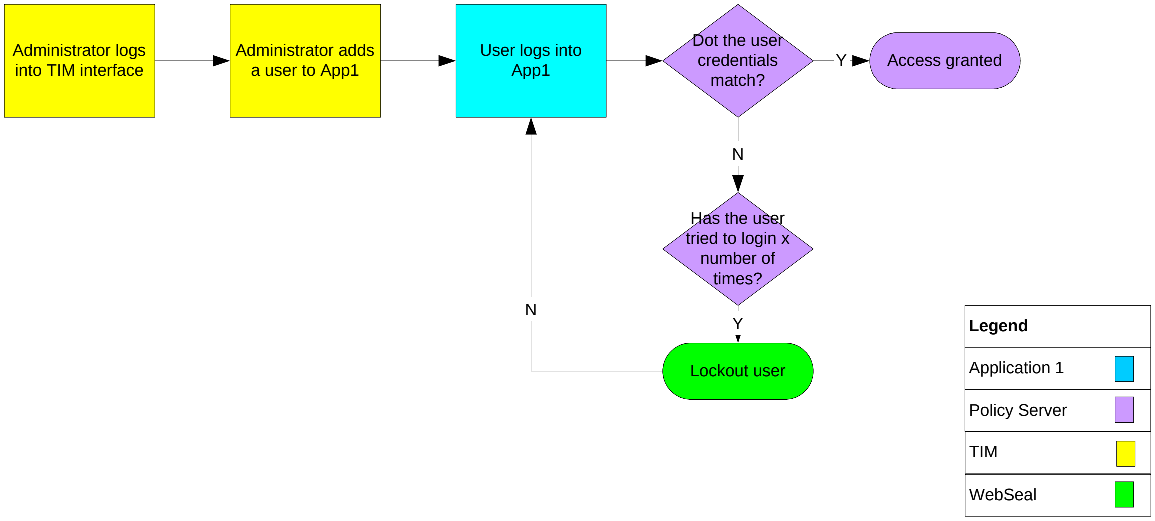

Administrator logs into TIM interface
Administrator adds a user to App1
User logs into App1
Dot the user credentials match?
Y
Access granted
N
Has the user tried to login x number of times?
Y
Lockout user
N
Legend
Application 1
Policy Server
TIM
WebSeal

Figure 20 –Centralized Administrator Testing Flow 1

Test 2: Adding user to application 1 and 2

In this scenario the administrator adds a user to two different applications and test case checks to make sure that the user can long into both applications. In this scenario Single Sign-on is assumed, so that the user is not challenged for credentials by application 2. This flow is described in figure 21.

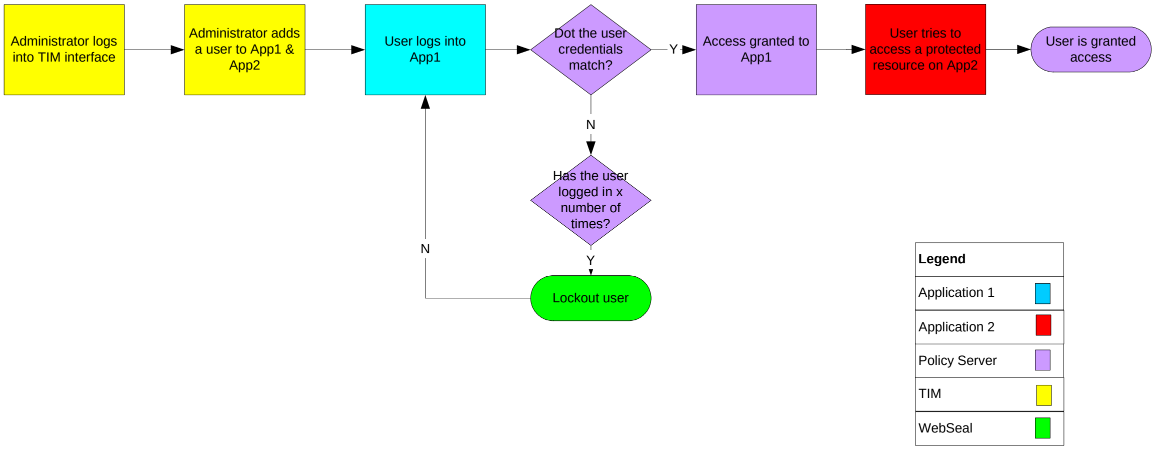

Administrator logs into TIM interface
Administrator adds a user to App1 & App2
User logs into App1
Dot the user credentials match?
Y
Access granted to App1
User tries to access a protected resource on App2
User is granted access
N
Has the user logged in x number of times?
Y
Lockout user
N
Legend
Application 1
Application 2
Policy Server
TIM
WebSeal

Figure 21 - Centralized Administrator Testing Flow 2

Test 3: Removing user from one application

In this test case, the centralized administrator removes user from application 2 but not application 1. The test is conducted to see that user is not able to access application 2 through TAM. This flow is shown in figure 22.

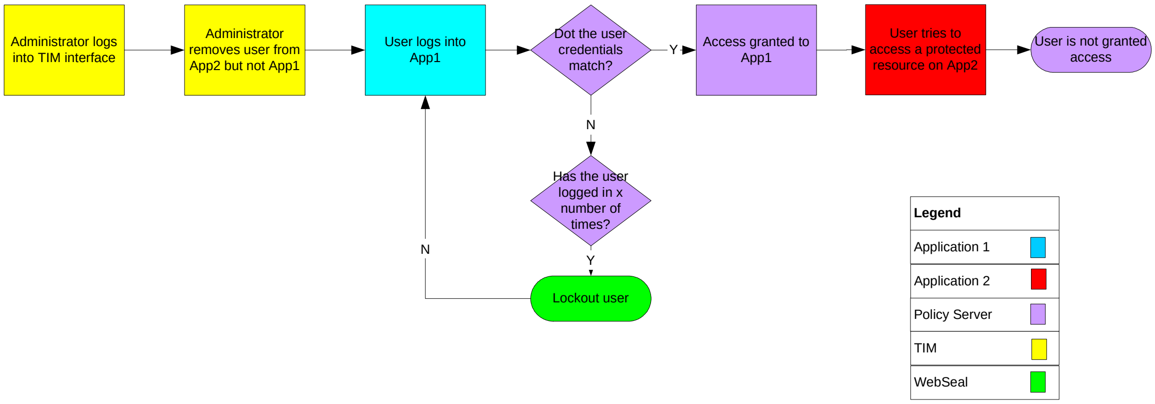

Administrator logs into TIM interface
Administrator removes user from App2 but not App1
User logs into App1
Dot the user credentials match?
Y
Access granted to App1
User tries to access a protected resource on App2
User is not granted access
N
Has the user logged in x number of times?
Y
Lockout user
N
Legend
Application 1
Application 2
Policy Server
TIM
WebSeal

Figure 22 - Centralized Administrator Testing Flow 3

Workflow Testing

The Proof of Concept included a simple workflow that would be tested to see how it can be implemented in a more complex environment. Note: This workflow was not tested due to the lack of a completely functioning TIM install. The testing flow is depicted in figure 23.

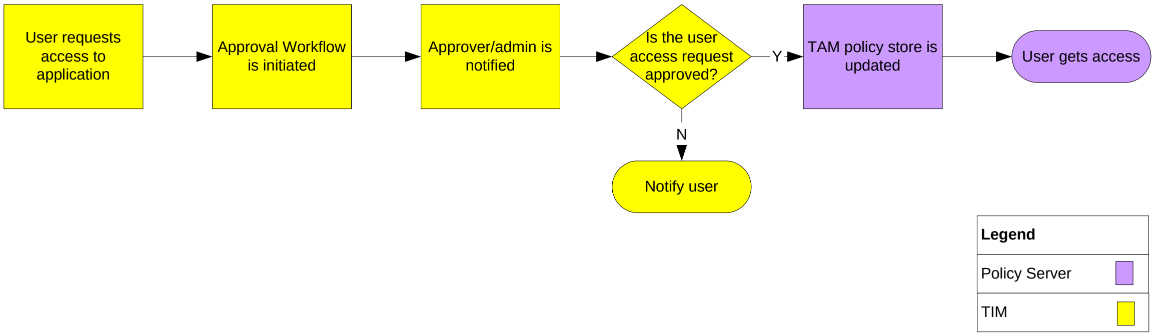

User requests access to application
Approval Workflow is initiated
Approver/admin is notified
Is the user access request approved?
Y
TAM policy store is updated
User gets access
N
Notify user
Legend
Policy Server
TIM

Figure 23 – Workflow Testing Flow

6.2 Testing Methodology

Testing methodology for TIM/TAM will involve testing the objectives and flows described in section 6.1. Security Tools Analysis team will create new users within the eZAudit and Expsite application. These user accounts will be used to test TIM/TAM test plan outline. This way no current user data will be exposed.

6.2.1 Tivoli Access Manager Testing Process

Testing process for TAM involved creating users with different access rights. The testing steps are described here.

Users and associated application access:

"ako01" – eZAudit & Experimental Sites
"sbarke01" – eZAudit only
"jkim" – Experimental Sites only
"asharma" - None

The following are example test cases. A complete list of test results can be found in appendix F.

Case 1: user "ako01"

1. "ako01" logs onto the WebSEAL portal and provides credentials
2. "ako01" selects the eZAudit application link from the portal
3. "ako01" is granted access to eZAudit
4. "ako01" types in the URL for Experimental Sites
5. "ako01" is automatically logged into Experimental Sites

Case 2: user "sbarke01"

1. "sbarke01" logs onto the WebSEAL portal and provides credentials
2. "sbarke01" selects the eZAudit application link from the portal
3. "sbarke01" is granted access to eZAudit
4. "sbarke01" types in the URL for Experimental Sites
5. "sbarke01" is not granted Expsite access

Case 3: user "jkim"

1. "jkim" logs onto the WebSEAL portal and provides credentials
2. "jkim" selects the Experimental Sites application link from the portal
3. "jkim" is granted access to Experimental Sites
4. "jkim" types in the URL for eZAudit
5. "jkim" is not granted eZAudit access

Case 4: user "asharma"

1. "asharma" logs onto the WebSEAL portal and provides credentials
2. "asharma" gets a message "not authorized user" error message

6.2.2 Tivoli Identity Manager Testing Process

TIM testing involved creating new user accounts, modifying user accounts, deleting user accounts, setting password policies and auditing administrator access. Once the TIM to TAM provisioning agent was setup, users created in TIM with an assigned group were able to access eZAudit and Experimental Sites applications through WebSEAL. For this testing scenario, the same users that were created in TAM were again created through TIM and provisioned to TAM.

User Scenarios in TIM
The following are example test cases. A complete list of test results can be found in appendix G.

Case 1: user “ako01”

1. User “ako01” is created in TIM by the administrator
2. User “ako01” is associated with a TAMService account
3. User “ako01” is added to eZAudit_admin and Expsite_admin groups
4. User “ako01” provisioning is approved by administrator in TIM
5. User “ako01” is created in TAM
6. User “ako01” has access to both eZAudit and Expsite applications

Case 2: user “sbarke01”

1. User “sbarke01” is created in TIM by the administrator
2. User “sbarke01” is associated with a TAMService account
3. User “sbarke01” is added to eZAudit_user group
4. User “sbarke01” provisioning is approved by administrator in TIM
5. User “sbarke01” is created in TAM
6. User “sbarke01” has access to eZAudit application

Case 3: user “jkim”

1. User “jkim” is created in TIM by the administrator
2. User “jkim” is associated with a TAMService account
3. User “jkim” is added to Expsite_user group
4. User “jkim” provisioning is approved by administrator in TIM
5. User “jkim” is created in TAM
6. User “jkim” has access to Experimental Sites application

Case 4: user “asharma”

1. User “asharma” is created in TIM by the administrator
2. User “asharma” is associated with a TAMService account
3. User “asharma” is not added to any group
4. User “asharma” provisioning is approved by administrator in TIM

5. User “asharma” is created in TAM
6. User “asharma” has access to no applications

6.3 Results Summary

6.3.1 Tivoli Access Manager Results

Tivoli Access Manager passed the majority of the testing conditions listed in Appendix D – Tivoli Access Manager Test Plan Outline. Basic authentication and Single Sign-on functions were tested with eZAudit and Experimental Sites applications. TAM tests for installation in the FSA VDC environment were not successful. These tests were conducted on a version of TAM that was installed on an IBM laptop. The test result details for Tivoli Access Manager Functionality are provided in Appendix E – Tivoli Access Manager Test results.

6.3.2 Tivoli Identity Manager Results

Tivoli Identity Manager testing was done on a version installed in the local Windows environment at the FSA UPC location. Testing included provisioning TAM, eZAudit and Experimental Sites by adding users through Tivoli Identity Manager. The details on TIM testing are listed in Appendix G – Tivoli Identity Manager Test Results.

7 Conclusion

The prototype phase provided FSA with an opportunity to gain experience with installation and integration of security tools for managing and administering access to FSA systems. FSA requested that the vendor install Tivoli Identity Manager and Tivoli Access Manager on the HP-UX 11i standard hardware configuration at the VDC. Several issues with Tivoli Access Manager and Tivoli Identity Manager running on HP-UX 11i prevented a complete installation during the on-site installation window. However, the prototype team continued the prototype effort by utilizing a pre-installed Windows version of Tivoli Access Manager provided by the vendor; TAM's functionality was successfully tested. The vendor also provided a VMWare version of Tivoli Identity Manager but the delay of working adapters to TAM or to add information to the target application Oracle database prevented significant testing of TIM.

7.1 Lessons Learned

The Identity and Access Management Tools prototype effort was a valuable opportunity to understand the technical and operational implications for deployment of Identity and Access Management COTS technologies. Some of the experience obtained during the project also has more general application to similar technology prototyping efforts in the FSA IT environment. The following suggestions and observations may help streamline future efforts to deploy Identity and Access Management technologies;

- *Consider creating a pre-development sandbox demonstration environment* - FSA could facilitate Proof of Concept installation and testing by creating a stand-alone sandbox environment to install and test prospective Commercial Off The Shelf (COTS) products. While the VDC environment is suitable for a traditional software development effort, the software development procedures currently in place impeded installation of COTS software packages. The VDC software deployment procedures provide an appropriately disciplined process for production environments and for software progressing through the FSA Software Life Cycle (SLC) during production deployment. However, the rigorous change control processes needed for the shared VDC environment severely restricted the ability to deploy and test COTS software. For example, not having the ability to directly access test servers in an expedited fashion resulted in substantial delays at several points in the prototyping effort and limited the time available for *ad hoc* testing or demonstration of the software being evaluated.
- *Ensure prospective software is tested and certified on hardware matching those in the test environment* – Given the issues experienced in deploying the proof of concept solution, steps should be taken to ensure the compatibility of I&AM products with the FSA VDC environment. Before any software is installed in the VDC shared development or production environments, the vendors should verify the fact that the software was tested and certified on hardware and operating system platforms that match the FSA software and hardware configuration. UNIX system installations are typically more complex than a Windows menu-

driven installation. FSA should plan additional time for testing, configuration changes, and maintenance updates when acquiring new products that will be deployed on FSA hardware. Vendors should also be aware that products cannot be installed in a completely isolated environment. Since HP-UX 11i is not always immediately supported by software makers when they release new product versions, security, patch and audit requirements for the VDC will need to be confirmed and tested. Testing the software in a clean lab environment is not a guarantee of a successful installation at the VDC. Tivoli Identity Manager and Tivoli Identity Manager depend on a number of middleware components, each with specific platform configuration requirements. If FSA requires software to be installed in a pure HP-UX 11i environment, design alternatives will need to be considered. Some Tivoli Identity Manager agents may not be available for the platforms supported at the VDC. This may require interaction with software and configurations that satisfy VDC monitoring and access standards.

- *Research ways to complete a remote installation* – FSA should investigate alternative methods to facilitate remote installation of software. Currently, the requirement that all access to the VDC HP-UX 11i environment utilize the SUDO utility will severely restrict or even prevent installation of test software in the VDC. It is not practical to determine all the commands that will be required in the course of an installation. For example, new COTS software often requires kernel changes or other modifications to the server configuration that cannot be accomplished through SUDO. Commands for error debugging and investigating issues common to test installation, but difficult to foresee completely, must be specifically requested to obtain access through SUDO. With the current SLA for adding new commands to a user's SUDO profile requiring two work days, efficient installation and testing of COTS software is impractical.
- *Be aware of and reserve adequate time for current VDC change control processes*– All requests for work at the VDC must be made through the Rational ECM Change Control Tool. It is necessary to get exact requirements to the VDC change control process as soon as possible in the schedule. While the VDC is very helpful in tackling requested actions, most requests to the VDC are managed under a two week service level agreement. During the Identity and Access Management Tools Analysis project, relatively minor schedule changes (such as those resulting from delays in the final product selection) significantly affected the ability to meet VDC targets for advance notice. Future efforts should additional time to the project schedule to account for the VDC lead times.
- *Successful integration is highly dependent on knowledge of the tools and availability of application resources*– The application integration effort was successful because the Proof of Concept team had access to timely, detailed technical information from resources with intimate knowledge of the target applications. It will be critical during future application integration efforts to coordinate closely with the system business owners and technical leads.
- *Continue to coordinate planning and deployment efforts between the various FSA stakeholders*- FSA operates a complex business and technical environment that required the participation of multiple internal and external parties this Proof of Concept effort. It will be important during future security architecture

development efforts to have a facilitator for the interactions required between vendors, contractors, and FSA technical and operational personnel.

- *A Proof of Concept is very valuable in understanding implications for future Security Tool efforts* - This Proof of Concept documented items that will be critical in facilitating future efforts with security tools at FSA. A Proof of Concept was a beneficial but low risk way of evaluating the suitability of products and technologies for FSA's environment. The Next Steps section that follows elaborates on the most critical points to consider in the next phases of this work.

7.2 Next Steps

The Identity and Access Management Tools Analysis was intended as the first step toward deployment of an enterprise FSA Security Architecture. Although the Identity and Access Management capabilities investigated in this effort do not comprise all the technical security controls defined in the FSA Security and Privacy Architecture Vision, they do provide a number of critical security services for FSA systems.

Deployment of Identity and Access Management services for use by FSA systems will require several additional steps in the future. This section outlines some of the approach options and recommendations for beginning development of architecture components to deploy Identity and Access Management services. Major next steps to promote deployment of a security architecture could include the following activities:

- Train FSA development and operations staff on Identity and Access Management products.
- Plan deployment of Identity and Access Management services
- Procure licenses for Identity and Access Management components
- Deploy Identity and Access Management infrastructure components
- Integrate FSA systems and applications with Identity and Access Management components
- Plan development of additional Security and Privacy Architecture components, e.g., data privacy services, patch management services, centralized encryption services

These proposed steps are discussed in the following sections.

7.2.1 Training

Identity and access management tools can implement a variety of security functions that benefit end users, security developers, system owners, and system security personnel. To take full advantage of capabilities, each of these various user groups will require specialized training. As a first step in developing FSA skills, Accenture has arranged technical training on Tivoli Access Manager for up to five FSA employees, a benefit of the IBM and Accenture strategic alliance. The training class is IBM's System Administrator training for Tivoli Access Manager and is scheduled for May 18, 2004 to May 21, 2004. Additional training is available from the vendor on the Tivoli Identity Management tool. Training classes are available for both security administrators (e.g., System Security Officers who would administer user access) and for developers and system administrators who would be responsible for installation and maintenance of the software.

7.2.2 Deployment Planning for Identity and Access Management Tools

Deployment planning for Identity and Access Management tools will need to address a variety of technical and business tasks. Outlined below are the high-level activities that will need to be included in a deployment planning effort. (These activities primarily treat development of enterprise systems that will provide security services to individual systems. Each existing or new FSA system will also need to plan and develop the components required to integrate with a FSA security architecture and allow use of the security services it provides.)

Security Infrastructure Development

- Create security infrastructure deployment plan for system integration
- Validate business requirements for identity management and access management
- Design enterprise-wide roles and authorization policy
- Deploy initial production security architecture components
  - Deploy access management System
  - Deploy identity management System
- Deploy additional security architecture phases
  - Deploy enterprise security infrastructure
  - Deploy scaled security infrastructure
- Deploy security architecture operational support components
  - Develop operational support plans
  - Plan and deliver operational training

Business System Support

- Develop system/application integration guidelines for access management
- Develop system/application integration guidelines for identity management
- Develop system/application integration support framework
- Develop FSA enterprise security role framework

Security Workflow Processes

- Define FSA security registration & approval workflow requirements
- Develop production deployment plan for workflow processes
- Design and validate security registration & approval workflow processes
- Configure and test workflow system support for workflow processes
- Develop security workflow support processes (call center, system administration)
- Deploy production pilot for security workflow process
- Deploy subsequent phases of security workflow processes

7.2.3 Procure Licenses, Deploy Identity and Access Management Tools, and Integrate FSA Systems

License procurement for Identity and Access Management tools should be planned to leverage licenses across existing and new FSA systems, including user populations served by all the systems. In the short term, user populations may include FSA employees, FSA contractors, and FSA trading partners (educational institutions, financial partners, guarantee agencies, federal and state agencies, etc.)

There are a variety of strategies FSA could pursue for deploying security architecture services and integrating them with new and existing FSA systems. Three potential approaches are described below, along with advantages and disadvantages for each approach.

7.2.3.1 Option 1: Expand the Identity and Access Management Prototype

Provide continued support of the Identity and Access Management prototype to allow demonstration of I&AM Security Architecture component capabilities, then use the prototype as a basis for deployment of an enterprise FSA Security Architecture. Major steps in this approach would be:

- Purchase a small number of TIM & TAM software licenses.
- Arrange for continued use of ITA hardware for TIM, TAM, associated components, and test applications.
- Continue integration of one or more sample FSA applications
- Develop scenarios that demonstrate user authentication, single sign-on, user auditing, centralized user access administration, user management workflow, auditing, password management, and delegated administration.
- Communicate Security Architecture capabilities to FSA business and CIO groups through workshops, presentations, and demonstrations.
- Develop standards and guidelines for integration of FSA applications with the FSA Security Architecture functions.

  Advantages:

  - Allows FSA to gain valuable experience with implementation and operation of security architecture components.
  - Allows deployment of Identity and Access Management capabilities that will enable an enterprise FSA security architecture.
  - Provides timely direction for in progress development efforts on FSA standards and guidelines for security development.

  Disadvantages:

  - Requires interim funding for hardware, software, and development of security components.

7.2.3.2 Option 2: Extend security architecture components provided by future system development efforts.

This approach would incorporate security architecture requirements in pending system development efforts, and leverage the security components provided by the new systems to establish an FSA Security Architecture. Major steps would include:

- Define requirements that can be used as infrastructure for enterprise deployment of FSA security architecture components and services.
- Include requirements for security architecture components in current system development specifications.
- Establish security architecture components deployed by systems currently in procurement as the standard for future FSA system development efforts.
- Extend security architecture components to provide enterprise security services across FSA systems.

Advantages:

- Eliminates need for current funding of security architecture efforts.
- Establishes FSA Security Architecture components as a standard for a major FSA system

Disadvantages:

- Poses risk of creating additional "siloed" FSA systems that have incompatible security architectures.
- May require additional costs if systems in development initially deploy security functions not suitable for integration with an FSA security architecture.
- Delays availability of FSA security services for other FSA systems.

7.2.3.3 Option 3: Require future migration of FSA systems to a standard security architecture

In this approach, new systems would be allowed to deploy security architecture components that meet their own functional and operational objectives, without specifying FSA security architecture requirements. At some point in the future, FSA could then migrate the deployed systems to an FSA Security Architecture. (Optionally, existing systems could be integrated with the FSA Security Architecture, depending on criticality and replacement schedule.) Major steps would include:

- Do not specify security architecture requirements for existing system development efforts, and allow contractors to select their own approach to providing security services.
- Develop FSA security architecture standards.
- Modify systems to comply with FSA security architecture standards.

Advantages:

- Potentially lower initial cost of system development.

- Avoids cost of support a standard FSA security architecture.

Disadvantages:

- Higher eventual costs to migrate new systems to a standard FSA security architecture
- Delays benefits of a standard FSA security architecture as identified in previous security architecture evaluations.
- Promotes "siloed" development of security services, leading to inconsistent security and privacy controls across different FSA systems.

7.2.4 Deploy Additional Security Architecture Components

The focus of this Task Order has been development of Identity and Access Management tools as a first step in the creation of standard FSA technical services provided by the FSA Security Architecture. However, the overall FSA Security and Privacy Technical Architecture defines a variety of other components that could be provided as services for FSA systems. For example, future efforts could target implementation of services for data privacy, patch management, and centralized control of data encryption.

Appendix A: Identity and Access Management Prototype VDC Installation Issues

The following list documents the results and issues identified during the VDC on-site installation of Tivoli Access Manager and Tivoli Identity Manager software during the week of May 3 – 7, 2004.

Individuals referenced in the table below:
RA – Rob Adachi (IBM)
SD – Scot Downs (IBM)
JH – Joe Hamblin (IBM)
SK – Srini Kankanahalli (FSA)

Date	Application	Result or Issue	Resolution
5/3/2004	TAM/TIM	Could not mount the windows burned CDRs (HSFS) on HP-UX 11i.	We had to mount the CDRs on a Windows computer and FTPed to the HP-UX 11i box. RA commented that there is another command to mount the CDRs under HP-UX 11i (as High Sierra CDFS)
5/3/2004	TAM/TIM	Could not unzip PKZIPed files under HP-UX 11i.	The unzip software had to be located and installed on the HP-UX 11i box.
5/3/2004	TAM	LDAP Server could not be started because of a JVM version conflict and environment configuration issue	JRE 1.4.2 had to be downloaded and installed on the HP-UX 11i boxes. The Java libraries path had to be updated to point to the 1.4.2 libs directory (the IBM software installation did not set the library paths correctly)
5/4/2004	TAM/TIM	Error trying to configure the policy server management tool. Error messages and warnings are shown when starting up the policy director manager stating that it cannot find the security certificate file. A ticket has been initiated with IBM support.	Pending. This appears to be a TAM bug. The contingency is to install the policy director on a windows pc to manage the LDAP server.
5/4/2004	TIM	DB2 doesn’t have the account/group set up in hp ux after running installation scripts.	All user/groups must be manually created prior to the installation of DB2
5/4/2004	TIM/TAM	Installation process is slow for components.	Problem may be due to virtual partitioning. Copying an 88 meg file from HPN1-TAM to HPN2 took over 1 hour. Not impacting the successful installation of systems so it has not been escalated.
5/4/2004	TIM	Upon installation, there were some users and files still left behind from the installation of WebSphere from the server image.	The IBM support told RA to go through a file/setup checklist to ensure all the files had been removed. No problems so far.

5/5/2004	TIM	DB2 does not start up when the required TIM patches are applied.	Backing out the patches allows DB2 to start up. RA will contact IBM support to see if the database issues are related to the DB2 config scripts.
5/6/2004	TIM/TAM	According to IBM, the machines that IBM certified TIM/TAM on for HP UX 11i are one year behind in the patch levels. This has caused the issues with TAM policy manager and TIM DB/2. IBM is running diagnostics to ensure they can mirror the machines.	IBM has procured HP UX 11i boxes and is attempting to mirror the VDC configuration for testing. This will involve developers and QA staff for TAM and TIM. SD and JH will also be working at an IBM lab in Raleigh, NC.
5/7/2004	TAM	IBM was unable to configure logging the for the policy director. They need to uncomment a log setting and then restart the policy director. The log files will then need to be sent to technicians for review.	We are waiting for IBM to send back the exact steps for this process so we can forward it to CSC.
5/7/2004	TAM/TIM	Because we are unable to install the software required for the Proof of Concept. IBM will provide a laptop with TAM installed and VM ware images for TIM.	IBM to provide laptop and VMWare image.
5/11/2004	TAM	IBM provided a list of commands and a ECM ticket was opened on behalf of IBM by SK.	Awaiting results from CSC
5/12/2004	TAM	CSC ran the requested commands and emailed the requested log to IBM	
5/17/2004	TAM/TIM	IBM successfully installed TIM and TAM on HP UX 11i servers that had the same patch levels as what was outlined in the server manifest. The engineers believe the problem may lie in the monitoring software installed at the VDC or software that was not successfully removed after the server imaging.	CSC has requested that IBM perform all additional work on these servers remotely.
5/19/2004	TAM	ECM ticket for the log files was closed.	

On 5/10, IBM provided a laptop with TAM installed. IBM provided CDs with TIM VM image but no connectors for TAM or Oracle were included. These components were requested. TAM agent was installed and configured but the Oracle agent was not fully configured in the given time frame.

Appendix B: Hardware Requirements and Configurations for Tivoli Access Manager and Tivoli Identity Manager

The following hardware requirements and configurations were provided by the vendor.

Tivoli Access Manager (TAM)

Requirements for installing TAM on the HP-UX 11i operating system.

Operating System Platform	Tivoli Access Manager 5.1 supported systems	Required Patches or Service Level
HP UX-11i	Authorization server Development (ADK) Java runtime environment Policy server Policy proxy server Runtime Web Portal Manager	PHCO_24400 v PHCO_24402 PHSS_25092 PHSS_26946 For specific languages only: Japanese:PHSS_26971 – Korean:PHSS_26973 – Simple-Chinese:PHSS_24975 – Traditional Chinese:PHSS_26977

TAM Disk Space and Memory Requirements

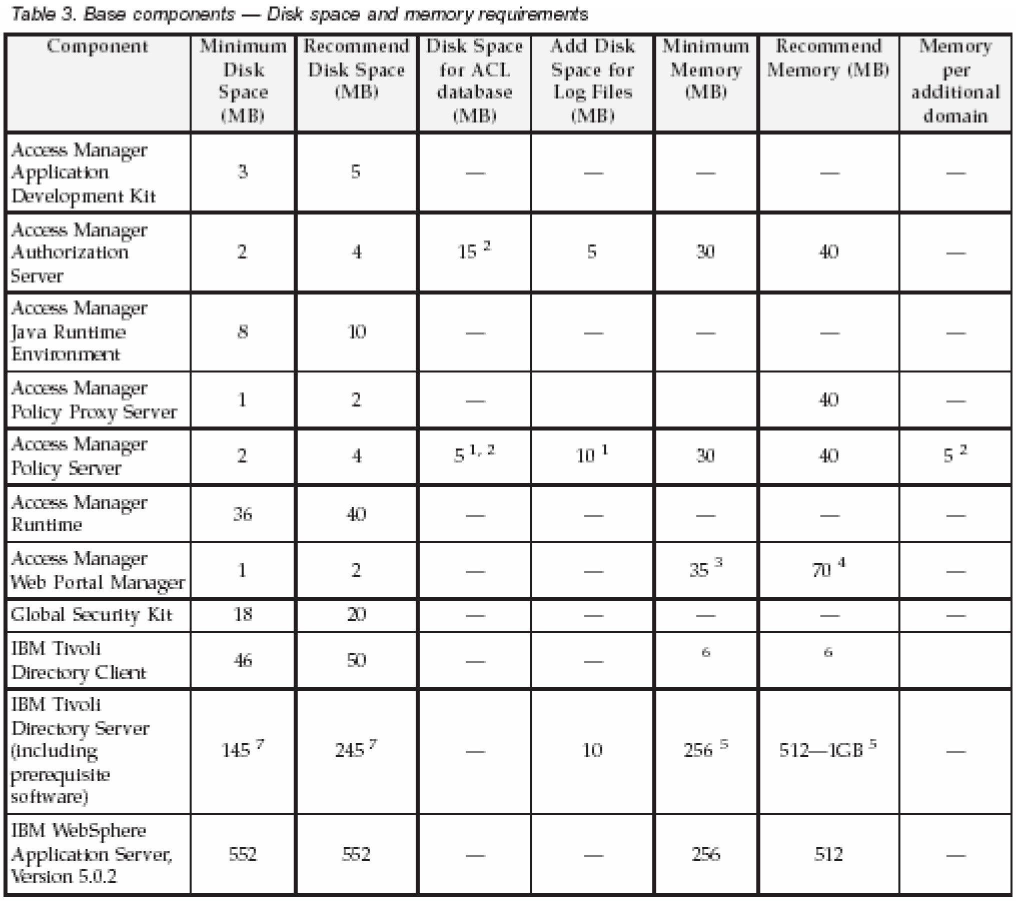


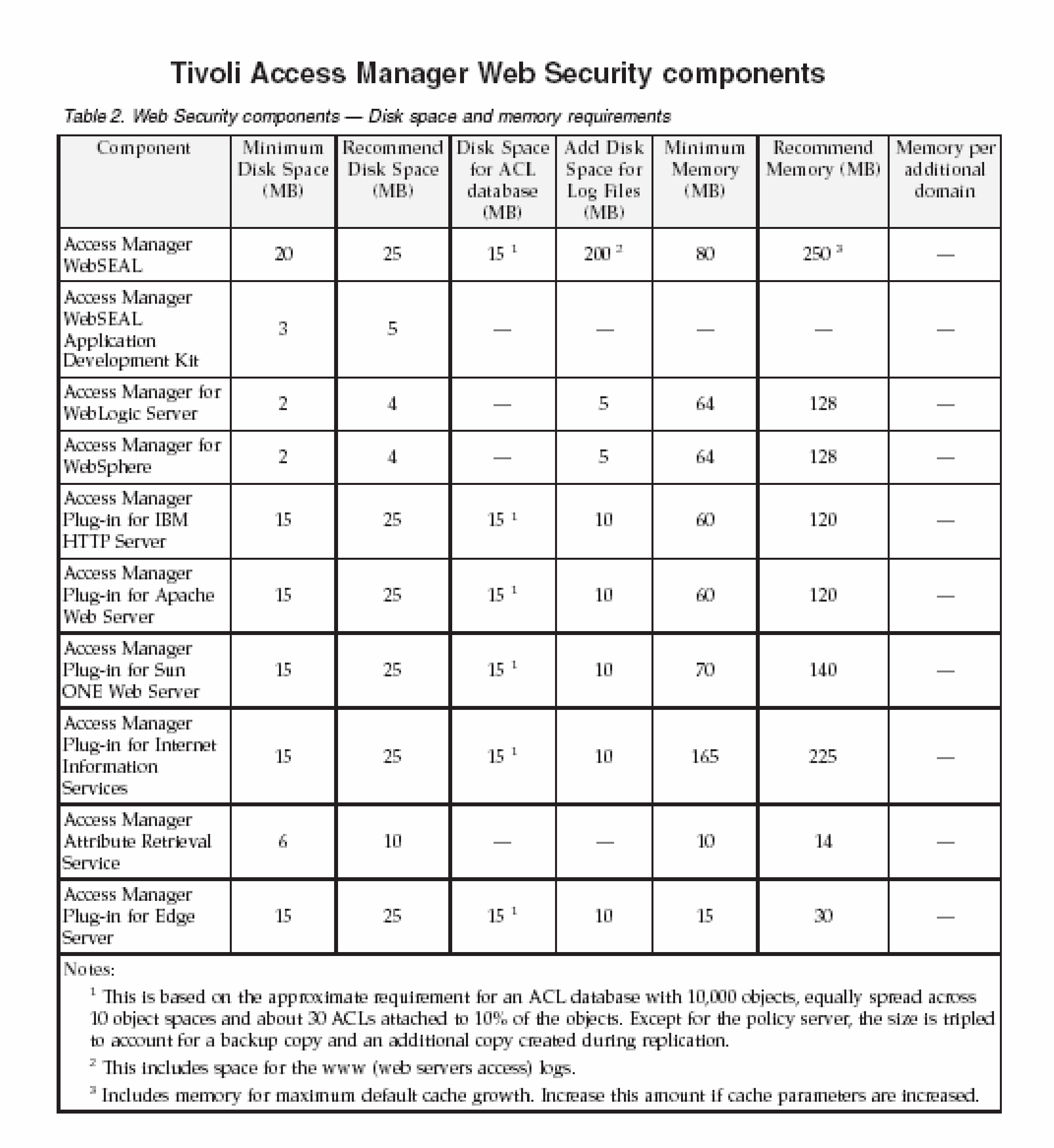
Table 3. Base components — Disk space and memory requirements

Component	Minimum Disk Space (MB)	Recommend Disk Space (MB)	Disk Space for ACL database (MB)	Add Disk Space for Log Files (MB)	Minimum Memory (MB)	Recommend Memory (MB)	Memory per additional domain
Access Manager Application Development Kit	3	5	—	—	—	—	—
Access Manager Authorization Server	2	4	15 [2]	5	30	40	—
Access Manager Java Runtime Environment	8	10	—	—	—	—	—
Access Manager Policy Proxy Server	1	2	—			40	—
Access Manager Policy Server	2	4	5 [1, 2]	10 [1]	30	40	5 [2]
Access Manager Runtime	36	40	—	—	—	—	—
Access Manager Web Portal Manager	1	2	—	—	35 [3]	70 [4]	—
Global Security Kit	18	20	—	—	—	—	—
IBM Tivoli Directory Client	46	50	—	—	[6]	[6]	
IBM Tivoli Directory Server (including prerequisite software)	145 [7]	245 [7]	—	10	256 [5]	512—1GB [5]	—
IBM WebSphere Application Server, Version 5.0.2	552	552	—	—	256	512	—

This information is taken from Tivoli Access Manager "Base Installation Guide". This guide can be accessed at http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business5.1.html

Reverse Proxy Server (WebSEAL) Requirements

Note: not all of the following components would be required for the FSA ITA environment.



Tivoli Access Manager Web Security components

Table 2. Web Security components — Disk space and memory requirements

Component	Minimum Disk Space (MB)	Recommend Disk Space (MB)	Disk Space for ACL database (MB)	Add Disk Space for Log Files (MB)	Minimum Memory (MB)	Recommend Memory (MB)	Memory per additional domain
Access Manager WebSEAL	20	25	15 [1]	200 [2]	80	250 [3]	—
Access Manager WebSEAL Application Development Kit	3	5	—	—	—	—	—
Access Manager for WebLogic Server	2	4	—	5	64	128	—
Access Manager for WebSphere	2	4	—	5	64	128	—
Access Manager Plug-in for IBM HTTP Server	15	25	15 [1]	10	60	120	—
Access Manager Plug-in for Apache Web Server	15	25	15 [1]	10	60	120	—
Access Manager Plug-in for Sun ONE Web Server	15	25	15 [1]	10	70	140	—
Access Manager Plug-in for Internet Information Services	15	25	15 [1]	10	165	225	—
Access Manager Attribute Retrieval Service	6	10	—	—	10	14	—
Access Manager Plug-in for Edge Server	15	25	15 [1]	10	15	30	—

Notes:

[1] This is based on the approximate requirement for an ACL database with 10,000 objects, equally spread across 10 object spaces and about 30 ACLs attached to 10% of the objects. Except for the policy server, the size is tripled to account for a backup copy and an additional copy created during replication.

[2] This includes space for the www (web servers access) logs.

[3] Includes memory for maximum default cache growth. Increase this amount if cache parameters are increased.

This information is taken from Tivoli Access Manager “Base Installation Guide”. This guide can be accessed at http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business5.1.html

Tivoli Identity Manager (TIM)

Requirements for installing TIM on the HP-UX 11i operating system.

Operating System Platform	Patch	Minimum memory, free disk space, and other hardware requirements
HP UX-11i	Quality Pack as of December 2002	RAM: 1 GB Processor: Clock speed of 440 MHz or faster Free disk space: /tmp must have 500 MB free disk space. Additionally, provide 150 MB for /itim45. {BEA_HOME} requires 300 MB of disk space.

This information is taken from Tivoli Identity Manager "Server Installation Guide on UNIX and Linux WebSphere". This guide can be accessed at http://publib.boulder.ibm.com/tividd/td/IdentityManager4.5.1.html

Notes:
1. HP-UX 11i: Tivoli Identity Manager has prerequisites for WebSphere Application Server and WebSphere embedded messaging support that require additional HP-UX 11i kernel settings. You must edit the /usr/conf/master.d/core-hpux file must be edited to allow the SAM utility to set values greater than 2048.

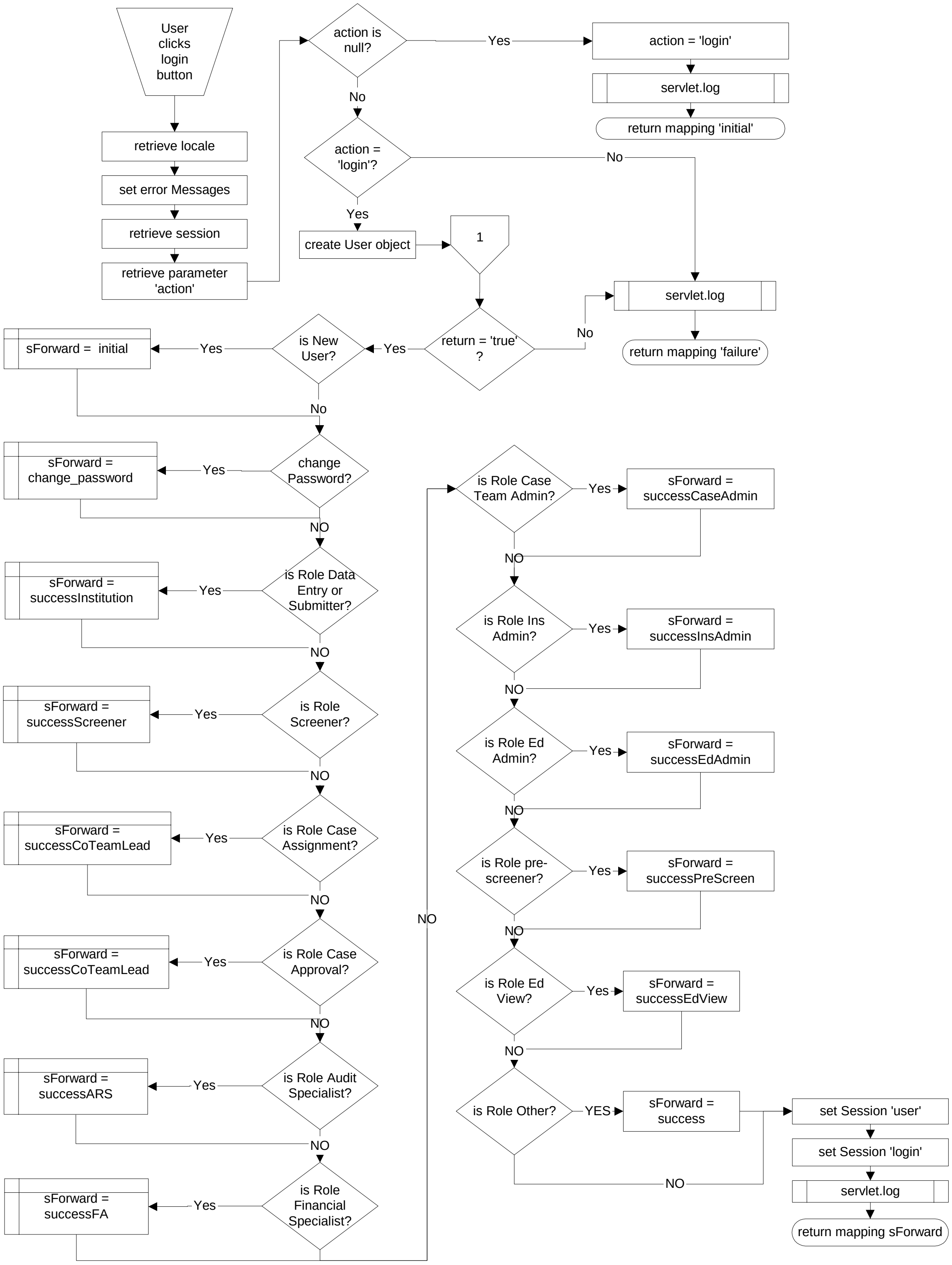
kernel values.

dbc_max_pct=25
maxdsiz=805306358
maxdsiz=2048000000 (when the base and Network Deployment products are on one computer)
maxfiles_lim=8196 (Change this one before maxfiles.)
maxfiles=8000
maxssiz=8388608
maxswapchunks=8192
max_thread_proc=3000
maxuprc=512
maxusers=512
msgmap=2048
msgmax=65535
msgmax=131070 (when the base and Network Deployment products are on one computer)
msgmnb=65535
msgmnb=131070 (when the base and Network Deployment products are on one computer)
msgmni=50

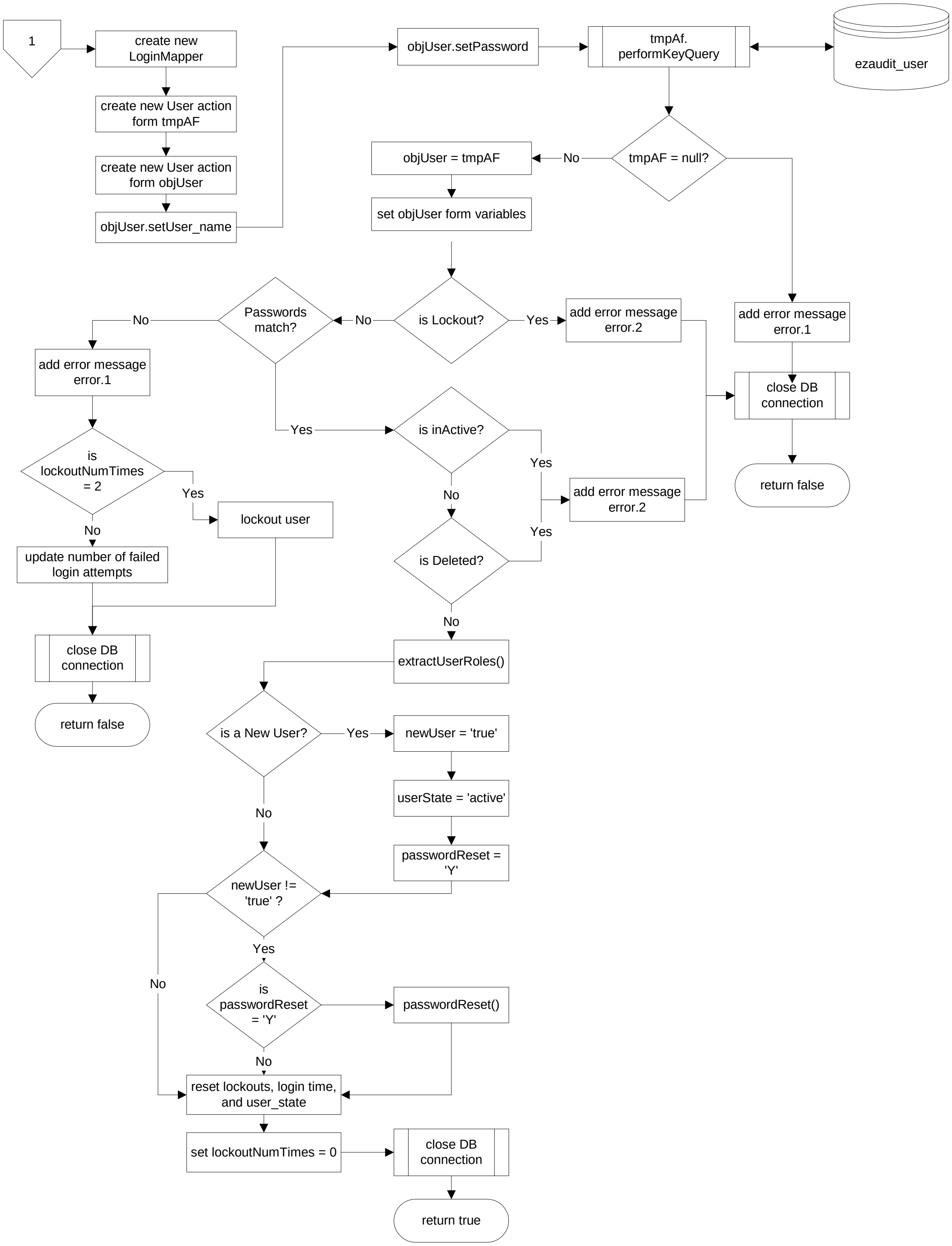
```
msgseg=32767
msgssz=32
msgtql=2046
nfile=58145
nflocks=3000
ninode=60000
nkthread=7219
nproc=4116
npty=2024
nstrpty=1024
nstrtel=60
sema=1
semaem=16384
semvmx=32767
semmap=514
semmni=2048
semmns=16384
semmnu=1024
semume=200
shmmax=2147483647
shmmni=1024
shmem=1
shmseg=1024
STRMSGSZ=65535
```

Appendix C: eZAudit Production and Prototype Logon Flows

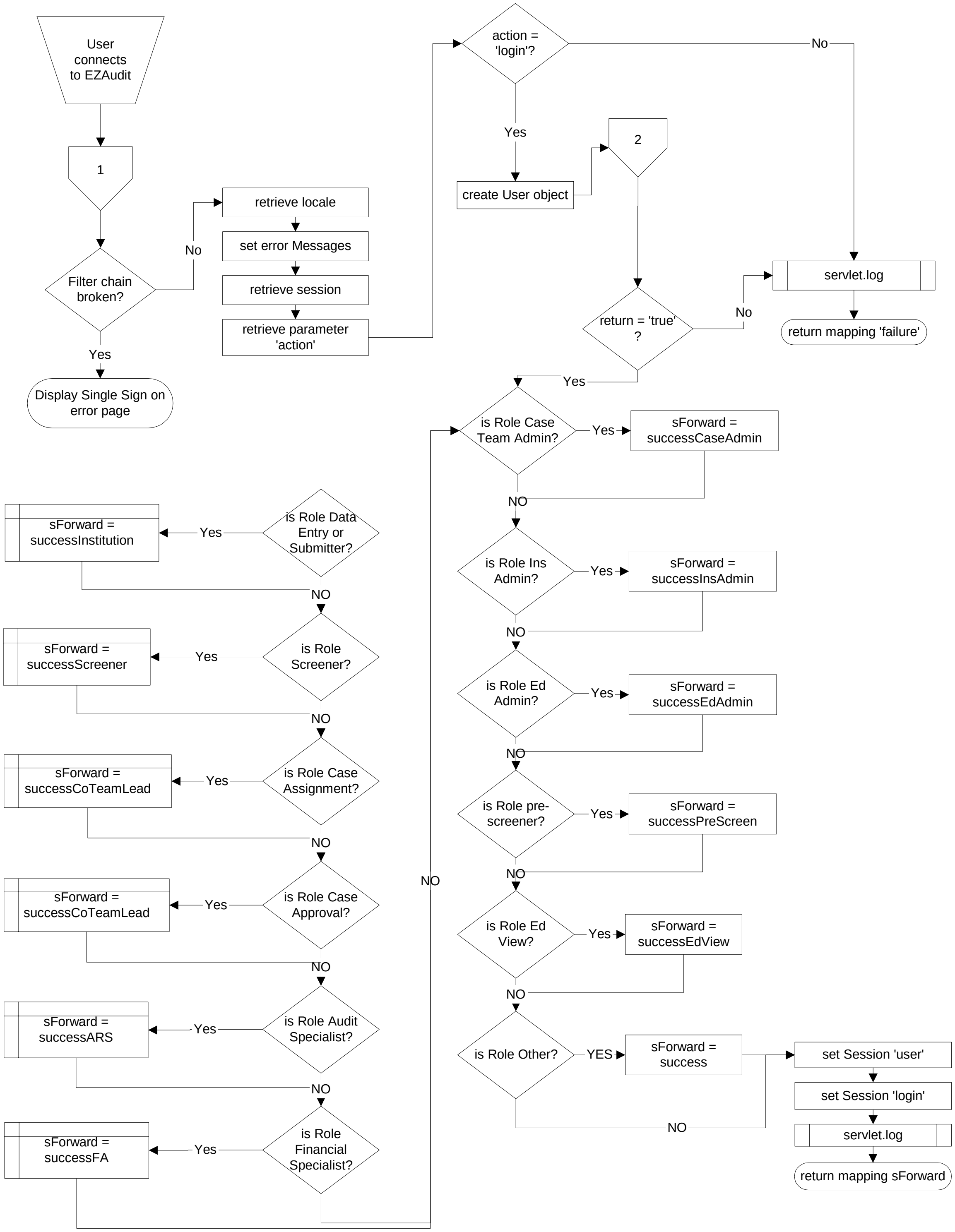
Production version (Logon Action)



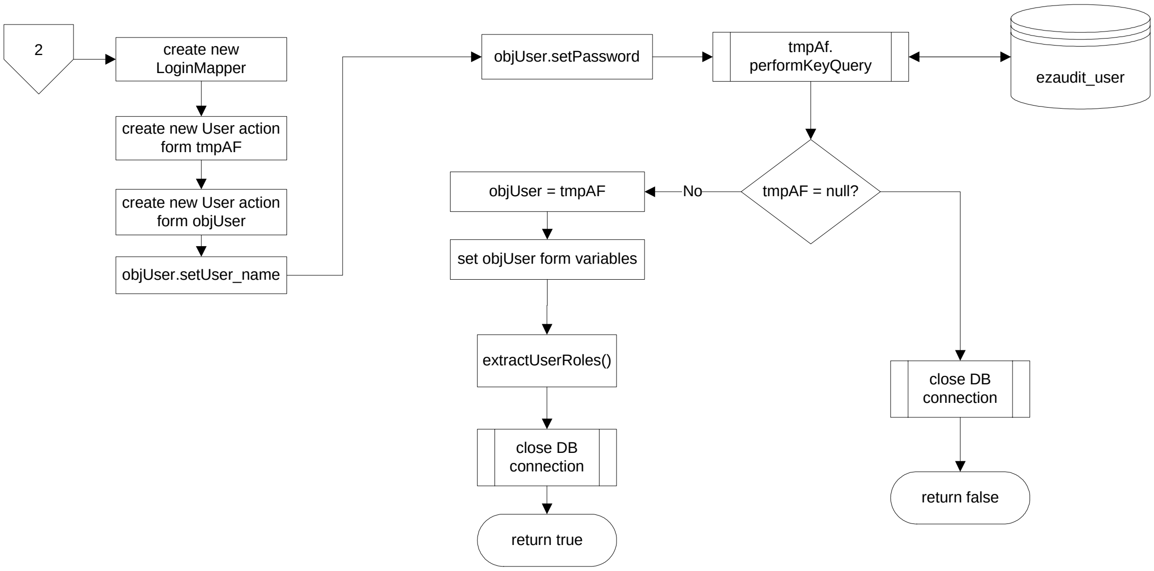
Production version (User Object)



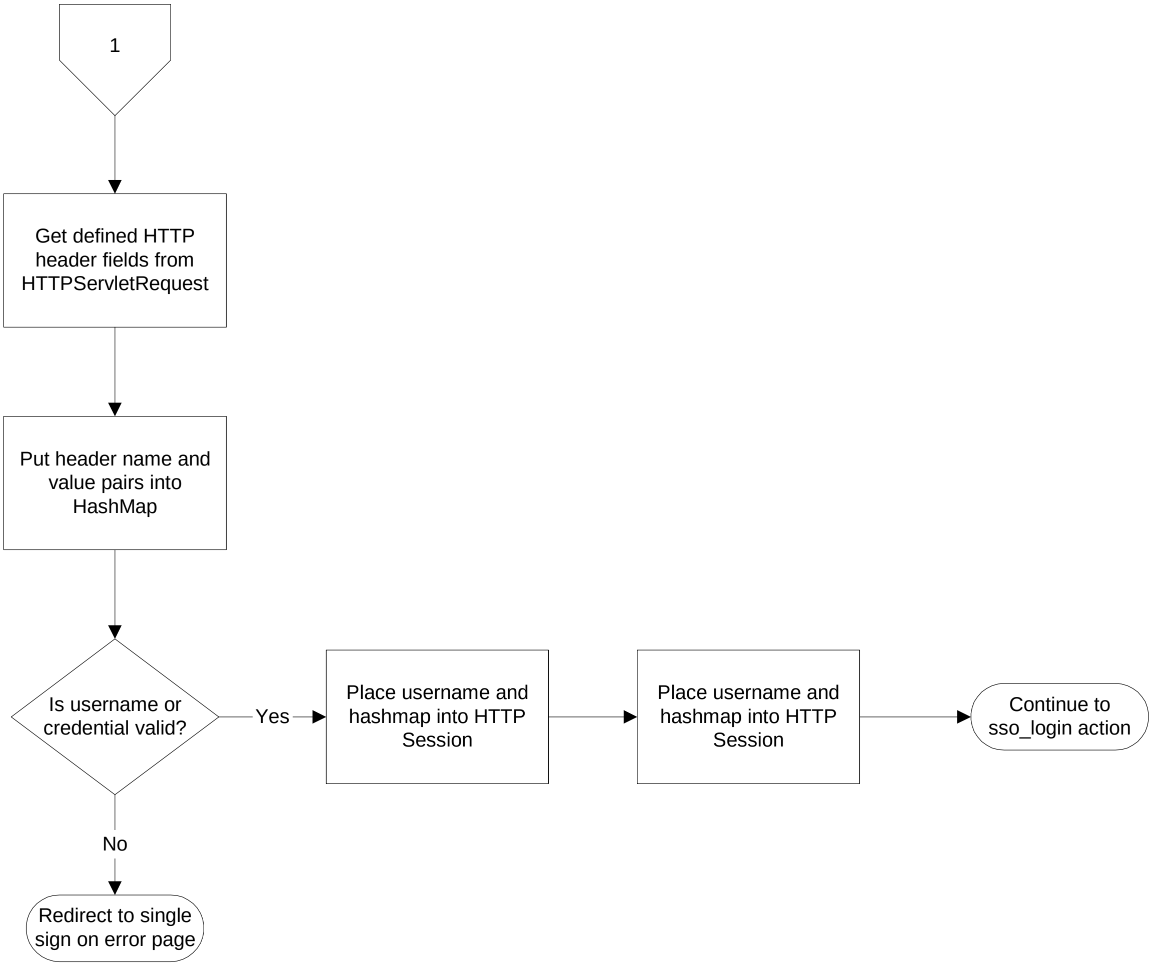
Proof of Concept (Single Sign-on Action)



Proof of Concept (User Object)



Proof of Concept (Authorization Filter)


1
Get defined HTTP header fields from HTTPServletRequest
Put header name and value pairs into HashMap
Is username or credential valid?
Yes
Place username and hashmap into HTTP Session
Place username and hashmap into HTTP Session
Continue to sso_login action
No
Redirect to single sign on error page

Appendix D: Tivoli Access Manager Test Plan Outline

Scenario	Success Criterion
1. Installation	
1.1 Installation of TAM	a) Installs without errors
1.2 Installation of WebSEAL	a) WebSEAL installs without errors
1.3 Establish communication between TAM and WebSEAL	a) Connectivity between TIM and WebSEAL can be verified
2. Testing Conditions	
2.1. User authentication	
2.1.1 Basic Conditions	
2.1.1.1. Authenticate a user logging into a protected application	a) Authorized user for gains access b) Unauthorized user is denied access
2.1.1.2 Accessing unprotected resources	a) User existing in TAM is allowed access to resource b) User not existing in TAM is allowed access to resource
2.1.1.3. Single-sign on functionality	a) Being authenticated to log into one application allows the user to log into any other application supporting single sign that the user is allowed to access b) Being authenticated to log into one application does not allow the user to log into any other application supporting single sign that the user is not allowed to access c) Being unauthenticated prompts authentication
2.1.1.4. Administrator revoked access	a) Revoking a user's access to an application in TIM denies them access to the application b) Revoking a user's access to one application TIM does not denies them access to the other application where they have access
2.1.1.5 Logging directly into a server	a) User's access to the application is denied.
2.2. Set and enforce security policies	
2.2.1 Basic Conditions	
2.2.1.1 Set or change lockout policy in TAM	a) User account is locked after X incorrect attempts
2.3. Audit	
2.3.1 Basic Conditions	
2.3.1.1. Generate audit report for user activity	a) For all Web applications b) For specific application c) For a specific user during defined period
2.3.1.2. Generate audit report for abnormal activity	a) Report authorization denials b) Report locked out users

Appendix E: Tivoli Identity Manager Test Plan Outline

Scenario	Success Criterion
1. Installation	
1.1 Installation of TIM	a) Installs without errors b) Application integration successful
1.2 Load existing user account information from eZAudit data store	a) eZAudit users are in TIM data store
1.3 Link user IDs from multiple platforms	a) Different User IDs linked to the same user
2. Testing Conditions	
2.1. Creating new accounts	
2.1.1 Basic Conditions	
2.1.1.1 Create a new user role	a) Role created, but no users assigned b) Assigned users have permissions specified
2.1.1.2 Add a new user	a) User created with no default role b) User created with specified role
2.1.2. Secondary Conditions	
2.1.2.1. Create new user with access provided in future	a) No user access until future date b) Date in past defaults to current date
2.1.2.2. Create new user with pre-defined expiration date	a) User has access at future date b) If past date specified, user has no access
2.1.2.3 Define delegated administrators	a) Delegated administrator setup
2.1.2.3. Limits on authority of delegated administrators	b) User can only be assigned to groups permitted
2.2. Modify user accounts	
2.2.1. Basic Conditions	
2.2.1.1. Modify user account attributes (e.g., first name, last name, e-mail etc)	a) Record is updated but access is not affected
2.2.1.2 Modify user role	a) Users assigned to the role have a new set of access permissions
2.2.1.3. Search for existing users	a) Search by full or partial user ID b) Search by full or partial user name
2.2.1.4. Administrator resets password from TIM	a) Password change propagates to the platform
2.2.2. Secondary Conditions	
2.2.2.1 User changes password	a) Password is updated in TIM but not in the applications
2.2.2.2 Assign time-limited access to a specific resource	a) Access to resource only until expiration date
2.3. Delete accounts or access	
2.3.1 Basic Conditions	
2.3.1.1. Delete a user account	a) User deleted from TIM, record retained
2.3.2 Secondary Conditions	
2.3.2.1. Partially revoke access permissions	a) User retains access to applications not deleted
2.3.2.2. Delete access for terminated employees	a) Access for terminated on specific date
2.3.2.3. Delete user at future date	a) User access expires at future date specified
2.4. Audit	
2.4.1. Generate audit report for assigned access	d) For all managed platforms e) For specific platform f) For a specific user during defined period

Scenario	Success Criterion
	g) Display access permissions for user
2.4.2. Audit administrator access	c) Report administrator actions d) Report administrator list e) Report administrator capabilities
2.5. Set and enforce security policies	
2.5.1. Basic Conditions	
2.5.1.1. Enforce password composition policy	a) Minimum password length b) Password composition set by platform c) Password composition set by TIM
2.5.1.2. Enforce password expiration policy	a) TIM forces password change
2.5.2 Secondary Conditions	
2.5.2.1. Set or change password composition policy in TIM	a) TIM can set policy in distributed platform
2.5.2.2. Set or change password expiration policy in TIM	a) TIM can set policy in distributed platform b) TIM can change policies
2.5.2.3. Enforce password reuse policy	a) TIM prevents reuse of password
2.5.2.4. Set or change password reuse policy in TIM	a) Password history list stored
2.5.2.5. Enforce inactive user account policy	a) Inactive user account detected by TIM
2.6. Workflow functions	
2.6.1 Secondary Conditions	
2.6.1.1. Workflow process	a) Administrators receive e-mail notification of workflow items

Appendix F: Tivoli Access Manager Test Results

Scenario	Success Criterion	Pass/Fail	Description
1. Installation			
1.1 Installation of TAM	b) Installs without errors	VDC/HP-UX – Fail FSA UCP/Windows – Pass IBM Test Lab/HP-UX - Pass	TAM was not fully installed on HP-UX11i at the VDC. As an alternative, IBM provided a pre-installed Windows version. IBM also installed TAM on HP-UX 11i at their test lab.
1.2 Installation of WebSEAL	b) WebSEAL installs without errors	VDC/HP-UX – Fail FSA UCP/Windows – Pass IBM Test Lab/HP-UX - Pass	WebSEAL was not fully installed on HP-UX11i at the VDC. As an alternative, IBM provided a pre-installed Windows version. IBM also installed WebSEAL on HP-UX 11i at their test lab
1.3 Establish communication between TAM and WebSEAL	b) Connectivity between TIM and WebSEAL can be verified	VDC/HP-UX – Fail FSA UCP/Windows – Pass IBM Test Lab/HP-UX - Pass	TAM and WebSEAL were not installed at VDC. As an alternative, IBM provided a pre-installed Windows version. IBM also installed TAM and WebSEAL on HP-UX 11i at their test lab .
2. Testing Conditions			
2.1. User authentication			
2.1.1 Basic Conditions			
2.1.1.1. Authenticate a user logging into a protected application	a) Authorized user gains access b) Unauthorized user is denied access	Pass Pass	Tested with user ako01 Tested with user asharma
2.1.1.2 Accessing unprotected resources	a) User existing in TAM is allowed access to resource c) User not existing in TAM is allowed access to resource	Pass Pass	To access the unprotected resources users will have to go directly to that page and not through the reverse proxy server.
2.1.1.3. Single-sign on functionality	a) Being authenticated to log into one application allows the user	Pass	Tested with user ako01

	to log into any other application supporting single sign that the user is allowed to access b) Being authenticated to log into one application does not allow the user to log into any other application supporting single sign that the user is not allowed to access c) Being unauthenticated prompts authentication	Pass Pass	Tested with users sbarke01 and jkim Tested with users ako01, sbarke01 and jkim
2.1.1.4. Administrator revoked access	c) Revoking a user's access to an application in TIM denies them access to the application d) Revoking a user's access to one application TIM does not denies them access to the other application where they have access	Pass Pass	This functionality was successfully tested This functionality was successfully tested
2.1.1.5 Logging directly into a server	b) User's access to the application is denied.	Pass	Tested on eZAudit application
2.2. Set and enforce security policies			
2.2.1 Basic Conditions			
2.2.1.1 Set or change lockout policy in TAM	a) User account is locked after 3 incorrect attempts	Pass	User account is locked out for 60 seconds. Tested with users ako01, sbarke01 and jkim.
2.3. Audit			
2.3.1 Basic Conditions			
2.3.1.1. Generate audit report for user activity	a) For all Web applications b) For specific application c) For a specific user during defined period	Pass Pass Pass	TAM generates a text file with all audit activity for applications and users. This data can be filtered to pull out specific applications and users activity for a given time period.

2.3.1.2. Generate audit logs for abnormal activity	a) Report authorization denials b) Report locked out users	Pass Pass	This data can be filtered from the TAM audit text file.

Appendix G: Tivoli Identity Manager Test Results

Scenario	Success Criterion	Pass/Fail	Description
1. Installation			
1.1 Installation of TIM	a) Installs without errors b) Application integration successful	VDC/HP-UX – Fail FSA UCP/Windows – Pass IBM Test Lab/HP-UX - Pass	TIM was not fully installed on HP-UX11i in the VDC environment. It was locally installed on a Windows box. IBM also successfully installed TIM on HP-UX 11i at their test lab.
1.2 Load existing user account information from eZAudit data store	a) eZAudit users are in TIM data store	Not Attempted	The TIM RBDMS agent required to complete this task has configuration issues. The same test using IDI was not attempted.
1.3 Link user IDs from multiple platforms	a) Different User IDs linked to the same user	Not Attempted	Task not completed in the limited time period
2. Testing Conditions			
2.1. Creating new accounts			
2.1.2 Basic Conditions			
2.1.2.1 Create a new user role	a) Groups created, but no users assigned b) Assigned users have permissions specified	Pass Pass	Groups can be created both in TIM and TAM User has access to specified resources
2.1.2.2 Add a new user	a) User created with no default role b) User created with specified role	Pass Pass	Users created in TIM Users created in TIM assigned roles
2.1.3. Secondary Conditions			

Scenario	Success Criterion	Pass/Fail	Description
2.1.2.1. Create new user with access provided in future	a) No user access until future date b) Date in past defaults to current date	Pass Pass	Created user will have access on a given future date
2.1.2.2. Create new user with pre-defined expiration date	a) User has access at future date b) If past date specified, user has no access	Pass Pass	TIM allows for users to be created with access on future date
2.1.2.4 Define delegated administrators	a) Delegated administrator setup	Pass	TIM allows for creation of delegated administrators
2.1.2.3. Limits on authority of delegated administrators	a) User can only be assigned to groups permitted	Pass	Delegated administrators in TIM can be assigned to administer only certain groups.
2.2. Modify user accounts			
2.2.3. Basic Conditions			
2.2.1.1. Modify user account attributes (e.g., first name, last name, e-mail etc)	a) Record is updated but access is not affected	Pass	User profile is updated
2.2.1.2 Modify user role	a) Users assigned to the role have a new set of access permissions	Pass	Changing user roles in TIM, changes access permissions
2.2.1.3. Search for existing users	c) Search by full or partial user ID d) Search by full or partial user name	Pass Pass	TIM allows for search functionality TIM allows for search functionality
2.2.1.4. Administrator resets password from TIM	a) Password change propagates to the platform	Not Attempted	This functionality was not tested as part of this prototype.
2.2.4. Secondary Conditions			
2.2.2.1 User changes password	a) Password is updated in TIM but not in the applications	Pass	Users can update passwords in TIM
2.2.2.2 Assign time-limited access to a specific resource	a) Access to resource only until expiration date	Not Attempted	This functionality was not tested in TIM, but is provided by TAM in password policy configuration screen.

Scenario	Success Criterion	Pass/Fail	Description
2.3. Delete accounts or access			
2.3.3 Basic Conditions			
2.3.1.1. Delete a user account	a) User deleted from TIM, record retained	Pass	TIM allows for this functionality
2.3.4 Secondary Conditions			
2.3.2.1. Partially revoke access permissions	a) User retains access to applications not deleted	Pass	Successfully tested
2.3.2.2. Delete access for terminated employees	a) Access for terminated on specific date	Pass	Successfully tested
2.3.2.3. Delete user at future date	a) User access expires at future date specified	Not Attempted	Standard TIM configuration does not allow this functionality.
2.4. Audit			
2.4.1. Generate audit report for assigned access	a) For all managed platforms b) For specific platform c) For a specific user during defined period d) Display access permissions for user	Pass Pass Pass Pass	TIM allows for reports to be generated in PDF format.
2.4.2. Audit administrator access	a) Report administrator actions b) Report administrator list c) Report administrator capabilities	Pass Pass Pass	TIM allows for reports to be generated in PDF format.
2.5. Set and enforce security policies			
2.5.2. Basic Conditions			
2.5.1.1. Enforce password composition policy	a) Minimum password length b) Password composition set by platform c) Password composition set by TIM	Pass Pass Pass	TIM can enforce the targeted platform password policy
2.5.1.2. Enforce password expiration policy	a) TIM forces password change	Pass	TIM enforces password expiration policy
2.5.3 Secondary Conditions			
2.5.2.1. Set or change password composition policy in	a) TIM can set policy in distributed	Not Attempted	Functionality not tested

Scenario	Success Criterion	Pass/Fail	Description
TIM	platform		
2.5.2.2. Set or change password expiration policy in TIM	a) TIM can set policy in distributed platform b) TIM can change policies	Not Attempted Not Attempted	This functionality was not tested as a part of this prototype Functionality not tested
2.5.2.3. Enforce password reuse policy	a) TIM prevents reuse of password	Pass	TIM password policy rules allow to set password reuse policy
2.5.2.4. Set or change password reuse policy in TIM	a) Password history list stored	Pass	TIM password policy configuration allows to set a password reuse policy and the password history is stored by TIM.
2.5.2.5. Enforce inactive user account policy	a) Inactive user account detected by TIM	Pass	TIM allows administrators to search for inactive accounts
2.6. Workflow functions			
2.6.2 Secondary Conditions			
2.6.1.1. Workflow process	a) Administrators receive e-mail notification of workflow items	Not Attempted	E-mail workflow not tested as part of this prototype

Appendix H: Code Examples for eZAudit Integration

Authorization Filter Code:

```
package gov.ed.fsa.security.authPOC;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.StringTokenizer;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * @author Anthony_C_Ko
 *
 * To change the template for this generated type comment go to
 * Window&gt;Preferences&gt;Java&gt;Code Generation&gt;Code and Comments
 */
public class FSAAuthFilter implements Filter {

        // resource bundle containing configuration info
        public static java.util.ResourceBundle m_filterProp;
        // hash map of values pulled out from HTTP header
        public Map m_headerValueMap;
        // list of the possible HTTP headers
        public static List m_headerList;
        private static String authFailureRedirectURL;


        /* (non-Javadoc)
         * @see javax.servlet.Filter#init(javax.servlet.FilterConfig)
         */

        public void init(FilterConfig arg0) throws ServletException {
                // get the configuration settings from the file
                m_filterProp = java.util.ResourceBundle.getBundle("authFilterConfig");
                StringBuffer sb = new StringBuffer();
                sb.append(m_filterProp.getString("credentialHeaders"));
                sb.append(",");
                sb.append(m_filterProp.getString("applicationHeaders"));
                authFailureRedirectURL = m_filterProp.getString("authFailureRedirectURL");
                StringTokenizer st = new StringTokenizer(sb.toString(), ",");
                ArrayList list = new ArrayList();

                m_headerList = list;

                //load the headers into the list;
                while(st.hasMoreTokens())
                {
                                String token = st.nextToken().trim();
                                list.add(token);
                                System.out.println(token);
                }

        }

        private Throwable handleException(Exception ex) throws ServletException
        {
                //TODO - handle logging
                throw new ServletException(ex);
        }

        /* (non-Javadoc)
         * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest,
javax.servlet.ServletResponse, javax.servlet.FilterChain)
         */
        public void doFilter(
                ServletRequest req_,
                ServletResponse res_,
                FilterChain chain_)
                throws IOException, ServletException {

                HashMap headerMap = new HashMap();
                m_headerValueMap = headerMap;
                boolean  authResult = false;

                if(req_ instanceof HttpServletRequest){
                        HttpServletRequest httpRequest = (HttpServletRequest) req_;
                        HttpSession httpSession = httpRequest.getSession();

                        //if there is an existing session, it should have a value.
                        if (httpSession != null)
                        {
                                boolean validSession = false;

                                String sessionToken =  (String)
httpSession.getAttribute("FSAAuthenticatedToken");
                                if(sessionToken != null && sessionToken.trim().equals("true"))
                                {
                                //TODO Make sure that the session is valid somehow
                                }
                                else{
                                        try{
                                                        for(Iterator iter =
m_headerList.iterator(); iter.hasNext(); )
                                                        {
                                                                String headerName = (String)
iter.next();
                                                                String headerValue =
httpRequest.getHeader(headerName);

        headerMap.put(headerName,headerValue);
                                                                System.out.println("Putting " +
headerName + " = " + headerValue);
                                                        }

                                                }
                                                catch (Exception e)
                                                {
                                                        handleException(e);
                                                }
                                                String userName = (String) headerMap.get("iv-
user");

                                                if(userName != null && userName.trim().length() >0
)
                                                {
                                                        //ensure the user name is valid

        httpSession.setAttribute("FSAAuthHeaderMap", headerMap);

        httpSession.setAttribute("FSAAuthenticatedUser",userName );

                                                        //we have passed the authorization check
                                                        authResult = true;
                                                }
                                        }
                                }
                                if(authResult ==false)
                                {
                                        HttpServletResponse httpResponse = (HttpServletResponse )
res_;
                                        httpResponse.sendRedirect( authFailureRedirectURL );
                                        return;
                                }
                        }


                chain_.doFilter(req_, res_);
        }

        /* (non-Javadoc)
         * @see javax.servlet.Filter#destroy()
         */
        public void destroy() {
                // TODO Auto-generated method stub
        }

        /**
         * @return
         */
        public static String getAuthFailureRedirectURL() {
                if (authFailureRedirectURL==null)
                {
                        return "not initialized";
                }

                return authFailureRedirectURL;
        }

        /**
         * @param string
         */
        public static void setAuthFailureRedirectURL(String string) {
                authFailureRedirectURL = string;
        }

}
```

Authorization Filter Config:

```
credentialHeaders=iv-user,iv-groups,iv-creds
applicationHeaders=
authFailureRedirectURL=common/sso_logout_error.jsp
iv-user=
iv-groups=groups
iv-creds=credentials
```

Code Fragment from eZAudit Login Action Class:

Code Fragment from eZAudit SSO enabled Login Action Class:

```
    if (user_id != null && user_id.length() > 0) {
                                        //[ako] changed code to get the user id from the HTTP
header instead of a form value
                                        UserBC bcUser = new UserBC(user_id);
                    String sForward;

                    //if the user does not exist in the database
                                        if(!bcUser.sso_login()){
                                                servlet.log(user_id + " not found in the
database");
                                                return (mapping.findForward("failure"));
                                        }
                                        else
                                        {
                                                if (bcUser.isRole("Data
Entry")||bcUser.isRole("Submitter")) {

                                                        sForward = "successInstitution";
                                                } else if (bcUser.isRole("Screener")) {
                                                        sForward = "successScreener";
                                                } else if (bcUser.isRole("Case Assignment")
                                                                  || bcUser.isRole("Case
Approval")) {

                                                        sForward = "successCoTeamLead";
                                                } else if (bcUser.isRole("Audit Specialist")){
                                                        sForward = "successARS";
                                                } else if (bcUser.isRole("Financial Specialist")){
                                                        sForward = "successFA";
                                                } else if (bcUser.isRole("Case Team Admin")){
                                                        sForward = "successCaseAdmin";
                                                } else if (bcUser.isRole("Institution Admin")){
                                                        sForward = "successInsAdmin";
                                                } else if (bcUser.isRole("ED Admin")){
                                                        sForward = "successEdAdmin";
                                                } else if (bcUser.isRole("Pre-Screener")){
                                                        sForward = "successPreScreen";
                                                } else if (bcUser.isRole("ED View")){
                                                        sForward = "successEdView";
                                                } else {
                                                        sForward = "success";
                                                }

                                                session.setAttribute("user", bcUser);
```

```
                                       session.setAttribute("login", "true");
                                       servlet.log("successful login");
                                       return (mapping.findForward(sForward));
                                       }
                               }

}
```

Appendix I: TAM & TIM HP-UX 11i Hardware Specs for IBM Test Lab Installation

These are the specs provided by IBM for HP UX 11i boxes used by them for their test lab TAM and TIM installation.

HP-UX 11i IBM Tivoli Access Manager Version 5.1

Hardware
HPj6750
36 gig disk space
512 meg of Memory
10/100 ethernet

Patches
PHCO_24400 //Note we used the lastest version of these patches as of early April 2004
PHCO_24402
PHSS_25092
PHSS_26946

HP-UX 11i IBM Tivoli Identity Manager Version 4.5.1

Hardware
c3750
36 gig disk space
1 gig of memory
10/100 ethernet

Patches

Quality Pack as of December 2002 or later versions. We did the test with the latest updates on the HP-UX 11i as of early April 2004

Configuration
Free disk space: – /tmp must have 1 GB free disk space. –
WebSphere Application Server, WAS_HOME must have 800 MB free disk space
/var must have 300 MB free disk space
Allocate 500 MB for ITIM_HOME.

```
dbc_max_pct=25
maxdsiz=805306358
maxfiles_lim=8196 (Change this one before maxfiles.)
maxfiles=8000
maxssiz=8388608
maxswapchunks=8192
max_thread_proc=3000
maxuprc=512
maxusers=512
msgmap=2048
msgmax=65535
msgmnb=65535
msgmni=50
msgseg=32767
msgssz=32
msgtql=2046
nfile=58145
nflocks=3000
ninode=60000
nkthread=7219
nproc=4116
npty=2024
nstrpty=1024
nstrtel=60
sema=1
semaem=16384
semvmx=32767
```

```
semmap=514
semmni=2048
semmns=16384
semmnu=1024
semume=200
shmmax=2147483647
shmmni=1024
shmem=1
shmseg=1024
STRMSGSZ=65535
```